Matlab » TimeVault » Образование » Rails

LXFDVD

LINUX FORMAT

Главное в мире linux

Сентябрь 2008 № 9 (109)

Семь дистрибутивов!

64 Studio » Linux Mint
OpenSUSE 11 » и другие
а также: Firefox 3

Обнови Eee!

Ваш маленький ПК может больше с. 64

SUSE: снова лидер?

Флагман Novell сражается за право стать вашей следующей настольной ОС

Сообщения об ошибках

Исправьте их сами! с. 41

!607130!821725!

Google Desktop

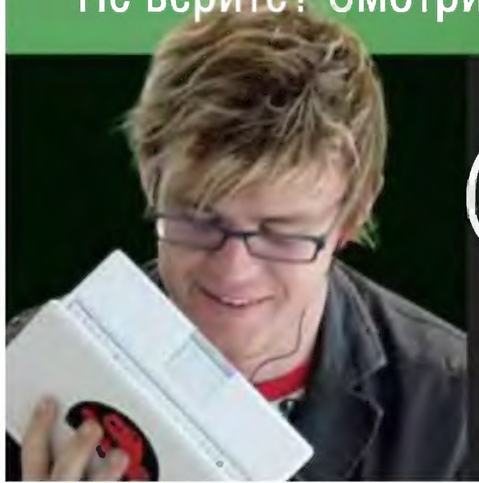
Linux + Google = нирвана.
Не верите? Смотрите сами! с. 60

Снова в школу

Linux в образовании:
взгляд из двух стран с. 104 и с. 106

Festival + Qt

GUI для чтения RSS – в буквальном смысле слова с. 76



“Наша миссия – отдать должное свободному ПО.”

Энди Фитцсаймон имеет ясную цель в жизни с. 49

Каталог агентства «РОСПЕЧАТЬ» – подписной индекс 20882
Каталог «ПРЕССА РОССИИ» – подписной индекс 87974

Linux Center
www.linuxcenter.ru

LINUX FORMAT
Главное в мире Linux

Оденем зеленые очки

В этом месяце мы уделяем повышенное внимание SUSE 11.0, поэтому и вопрос к нашим писателям был такой: «Если бы SUSE отказалась от талисмана-геккона, то что бы вы предложили взамен?»

Грэм Моррисон
Думаю, Робин-Бобин-Барабек подойдет как нельзя лучше.

Майк Сондерс
Педантичное замечание: логотип SUSE – не геккон, а хамелеон. Читайте Википедию!

Ник Вейч
У меня есть фотка Майка, пробующего на язык файлы. Окрасить в зеленые тона – будет самое то.

Эфраин Эрнандес-Мендоса
Пиво. Стойте, это был вопрос? Ну, что-нибудь немецкое... Скажем, пиво?

Эндрью Грегори
ASCII-портрет Дэвида Хесселхоффа. А все стандартные мелодии надо заменить на Wind Of Change.

Энди Ченнел
Черного дружка панды Су. Да, название тоже придется поменять.

Дэвид Картрайт
До тех пор, пока его компилируют в соответствии с требованиями Reinheitsgebot 1516 года, я буду счастлив.

Д-р Крис Браун
Разумеется, хамелеон выбран талисманом за свое умение мгновенно менять внешний вид? Как KDE...

Майкл Дж Хэммел
Лайм. Он тоже зеленый и совсем не лимон.

Дэниел Джеймс
Улыбающегося Стива Баллмера в бейсболке Novell.

Сьюзан Линтон
Тиранозавра, потому что SUSE – это мега-дистрибутив, сотрясающий землю Линуксландии, и внушающий всем уважение и страх.

Саймон Пикстон
Я бы оставил геккона, но сделал его из ливерной колбасы, квашеной капусты и картофельного пюре.

Миссия журнала

- Пропаганда свободного ПО в России
- Продвижение решений с открытым кодом в бизнес-сообществе
- Поддержка российского Open Source сообщества
- Организация трибуны для разработчиков свободного ПО
- Обратная связь между разработчиками и потребителями ПО

Семь «бед» – один ответ

Традиционно сентябрьский номер журнала открывался подведением итогов очередного года существования **LXF**. Так было бы и на сей раз, не перехвати у меня инициативу наш давний друг и соратник Алексей Федорчук. Получилось, на мой взгляд, очень даже интересно, а я сегодня буду говорить совсем о другом годе – об учебном.

Сейчас много рассуждают о продвижении свободного ПО в образовании. Тематика школьного Linux на страницах нашего журнала не нова: первый материал на этот счет был опубликован еще в LXF91, а регулярная рубрика появилась в LXF92. Но такого, как в LXF109, не случалось, пожалуй, ни разу. На страницах одного номера собран опыт людей, продвигавших открытую ОС сразу в трех странах: России, Украины, Великобритании. Плюс мнение эксперта, считающего, что Linux уже созрел для большого бизнеса, но не ставящий стоимость во главу угла. Разные учреждения, разная мотивация, но достаточно близкие результаты. Наверное, этого и следовало ожидать: ведь если полученный ответ к задаче является правильным, он не должен зависеть от выбранного способа решения. А может быть, не такие уж мы все и разные... **LXF**

Валентин Синицын » Главный редактор info@linuxformat.ru

Как с нами связаться

Письма для публикации: letters@linuxformat.ru
Подписка и предыдущие номера: subscribe@linuxformat.ru
Техническая поддержка: answers@linuxformat.ru
Проблемы с дисками: disks@linuxformat.ru
Общие вопросы: info@linuxformat.ru
Web-сайт: www.linuxformat.ru

- **Адрес редакции:** Россия, Санкт-Петербург, Лиговский пр. 50 корп. 15
- **Телефон редакции:** (812) 640-49-90. Дополнительная информация на стр.142

LINUX FORMAT

Содержание

Весь номер – прямо как на ладони: приятного чтения!

Обзоры

Penumbra: Black Plague 10
Бросайте SuperTux: в городе появились зомби-мозгоеды!

Qnap TS-409 Pro Turbo 12
В похожем на колонку корпусе скрывается настоящий NAS с RAID, способный вместить до 4 ТБ ~~мусора~~ ценных файлов.

› **Сюда можно скопировать 512 дисков LXF – на ближайшие лет сорок места вам хватит.**

JBuilder 2008 Enterprise 13
Свобода не дает даром – в этом случае за Java IDE на платформе Eclipse придется выложить почти 40 тысяч рублей.

CrossOver Linux 7 14
Нужно запустить Microsoft Office 2007 в Linux? Начните с CrossOver Linux 7!

Blender 15
Графический пакет промышленного уровня стал еще промышленнее. Чего не скажешь про интерфейс...

Сравнение: числодробилки

Euler Math Toolbox 19
Matlab 17
Octave 18
Sage 19
Scilab 20

Что за штука...

TimeVault

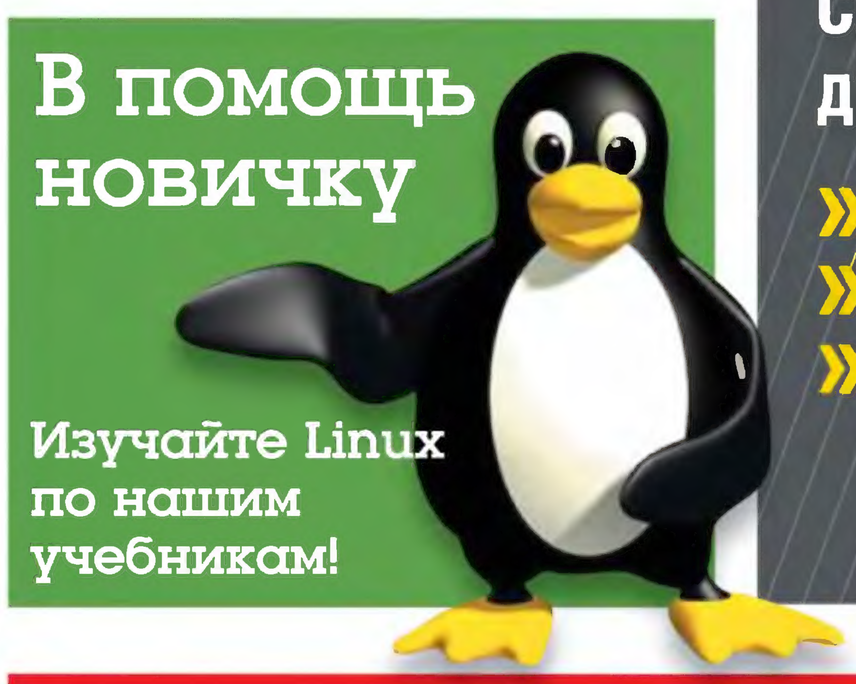
Система резервирования, навеянная мыслями о будущих поколениях с. 53

Почему Linux готов войти в большой бизнес с. 37

Люди говорят

> Наша миссия – отдать должное свободному ПО и объяснить про него людям

Энди Фитцсаймон – хакер для нехакеров с. 49



Сила Ubuntu в новой упаковке плюс дополнительное ПО

- Быстрые ядро и рабочий стол
- Простое управление пакетами
- Запускается прямо с DVD!

Спецрепортаж

Сообщения об ошибках 41
Не дайте им испортить вашу жизнь – смотрите, понимайте, исправляйте!

Libre graphics 49
Что происходит в мире свободного графического ПО – взгляд из Великобритании.

Учимся музицировать 45
Нам надоело ждать, пока школы во всем мире установят себе Linux, поэтому мы пришли в одну из них и показали, как это делается.

Подпишись на Linux Format и сэкономь!

Постоянные рубрики

Новости 4
События мира Linux глазами наших экспертов.

Distrowatch 22
В этом месяце мы восхищаемся SUSE 11, Zenwalk 5.2 и Linux Mint.

Интервью Linux Format 24
Фонд социального страхования РФ выбирает открытое ПО

История успеха 28
Linux в правовом бизнес-бюро Санкт-Петербурга

Что за штука 53
TimeVault: это как будто капсула времени, но со снимком файловой системы внутри.

Рубрика для админа 55
Ребята, вы не одиноки! Д-р Крис Браун разделяет вашу боль и стремится сделать мир чуточку лучше.

Игрострой 100
Эти загадочные шейдеры...

Школа LXF 104
Наша образовательная рубрика.

Ответы 110
Проблемы Linux решены: загрузка (Grub и Lilo) – 3 шт., беспроводная сеть – 1 шт., ну и по мелочи: установка ПО, CUPS и SD-карты.

LXFHotpicks 122
Горячие новинки мира Open Source.

Диск Linux Format 116
Live-версия SUSE 11 в редакции LXF

Через месяц 128
Ждать осталось совсем немного!

Этот ленивый козел [Indolent Ibex] вчера весь день валялся под окнами нашего офиса.

Учебники

Начинающим

Flock 60
Убивайте время вдвое быстрее с помощью Flock, настоящего обозревателя Web 2.0, или присоединяйтесь к революции онлайн-приложений с Google Desktop.

Любителям ноутбуков

Взломайте свой Eee 64
Выйдите за рамки Xandros и установите на микроноутбук другой дистрибутив по своему вкусу!

GIMP

Записки из камеры 68
GIMP – прекрасный растровый редактор, но знали ли вы, что его можно использовать и для работы с цифровыми фотоаппаратами?

Сам себе режиссер

Видеопереходы 88
Не утомляйте своего зрителя скачущей с одного на другое камерой – добавьте мягкое затухание или что-нибудь еще в этом роде.

KDE 4

Напишем плазмоид 80
Самая броская технология KDE 4 выглядит изнутри совсем не такой страшной – особенно после прочтения этого учебника.

CMake

Соберем проект 92
Если вы всерьез решили взяться за разработку для KDE 4, то без этих знаний вам не обойтись.

Scilab

Пакеты расширений 84
Узнайте, что еще умеет Scilab, загляните за горизонт с версией 5.0 и познакомьтесь с еще одной достойной программой.

R

Две переменные 96
Анализ данных продолжается! В этом выпуске: проверка гипотез нормальности и однородности.

Ruby on Rails

Flickr на коленке 72
В прошлом месяце мы набросали web-галерею, а сегодня добавим ей крутизны при посредстве Ajax и прочего Web 2.0.

Hardcore Linux

Python и Qt 76
Синтезатор речи из проекта прошлого месяца обзаводится приятным Qt-интерфейсом, доступным даже новичку.

KDE 4.1 – возможность выбора

Одно из самых заметных событий лета – выход новой версии популярного открытого рабочего стола KDE, 4.1. Некоторые специалисты уже назвали ее первой по-настоящему готовой редакцией данной среды. Давайте попробуем разобраться: правы ли они в своих оценках?

29 июля 2008 года проект KDE выпустил KDE 4.1. Это второй крупный релиз – первым был KDE 4.0, предназначенный в основном для разработчиков. Напомним, что четвертая версия популярного рабочего стола была написана практически «с нуля». В стабильной ветке появились новые приложения и возможности для разработчиков. Активно создается кроссплатформенный каркас – идет портирование приложений и библиотек в Windows и Mac OS X. Дирк Мюллер [Dirk Müller], один из релиз-менеджеров, приводит следующие числа: при разработке KDE 4.1 было внесено 20803 изменений в исходные тексты KDE 4.0, а в рабочих ветках – около 35000 изменений; некоторые из них также вошли в KDE 4.1. Однако мы не будем подробно описывать вещи, интересные только разработчикам. Если стоит вопрос о готовности среды, посмотрим: что изменилось для простого пользователя?

Согласно опубликованному пресс-релизу, появившаяся в KDE 4.0 Plasma «повзрослела», и теперь с ней могут работать даже новички. Рабочий стол реализован в виде приложения Plasma, и пользователь сможет вывести на него содержимое любого каталога. 4.1 – первый релиз KDE 4, в который вошел пакет KDE *PIM*, включающий *KMail*, *KNode*, *KOrganizer*, *Akregator* и другие приложения, интегрированные в оболочку *Kontact*. Появился видеоплейер (*Dragon Player*), а также был «реабилитирован» проигрыватель AudioCD (*KSCD*). Огромное количество исправлений и улучшений внесли в web-браузер *Konqueror*, менеджер файлов *Dolphin* и просмотрщик *Gwenview*. Композитный оконный менеджер *KWin* обзавелся новыми возможностями и стал работать стабильнее. Разумеется, список изменений можно продолжить.

Однако пока не ясно главное – готов ли KDE 4 для рабочего стола обычного пользователя? На этот вопрос очень сложно дать однозначный ответ. Конечно, темпы разработки впечатляют, а та сборка KDE 4.1 (из неофициального репозитория для Ubuntu 8.04), которую нам удалось «потрогать», выглядит стабильной. С другой стороны, существует ряд проблем, затрудняющих установку KDE 4.1 на компьютер «рядового пользователя».

Новая версия KDE пока не включена в стабильные ветки большинства популярных дистрибутивов. Разумеется, существуют неофициальные сборки, а также нестабильные и тестируемые репозитории – способ установки вы можете уточнить на сайте любимого варианта Linux или на соответствующих форумах. Хотелось бы особо отметить OpenSUSE: здесь можно установить KDE 4.1.0 буквально «одним кликом».

Итак, с инсталляцией разобрались. Теперь перейдем к главной сложности: на сегодняшний день переписаны далеко не все популярные приложения. Для KDE 4 нет, например, стабильных версий *KOffice*, *DigiKam*, *K3B* и *Amarok*. Конечно, никто не мешает вам запускать старые версии любимых программ, но тогда какой смысл менять окружение? Не проще ли пока пользоваться «вылизанным» KDE 3.5.9? Не будем также забывать об ошибке в двоичных драйверах NVIDIA (в основном она проявляется с адаптерами NVIDIA 8xxx, 9xxx, 2x0GTX), из-за которой в KDE 4 очень медленно перерисовывается интерфейс. Разработчики KDE здесь не виноваты, но это будет слабым утешением «счастливому» владельцу. Последняя проблема касается русскоязычных пользователей – локализация KDE 4 еще далека от идеала.

Подведем итог. Те, кто любит возиться с новыми программами, могут установить KDE 4.1 уже сейчас. Пользователям, которым нужен стабильный рабочий стол, стоит подождать, пока выйдет релиз любимого дистрибутива с включенным в него KDE 4. К тому времени и сам KDE доработают, и приложения под него перепишут.

Осталось решить еще один важный вопрос: насколько удачной была идея такой серьезной переработки KDE? Не лучше ли было вносить изменения постепенно, без глобальных потрясений? Нам кажется, что лидеры проекта KDE были правы, и современный открытый рабочий стол, ориентированный на простого пользователя – это именно то, что сейчас нужно. И незачем при этом тянуть совместимость с разработками прошлого века.

Те же, кого такой подход не устраивает, всегда могут перейти на другой рабочий стол или оконный менеджер. В этом и состоит главное преимущество мира Open Source – многообразие различных решений дает пользователю возможность выбора.

Новости короткой строкой

- Патентное бюро США планирует изменить политику патентов в области ПО. Существующие патенты могут быть отозваны; кроме того, выдачу новых планируют существенно ограничить.
- НАТО включает ISO/IEC 26300:2006 (OpenDocument) в список стандартов обмена информацией.
- O'Reilly Media опубликовала отчёт, согласно которому от 5% до 15% вакансий в сфере IT требуют навыков разработки открытого программного обеспечения.
- Компания Dell начинает поставки ноутбуков с предустановленным Ubuntu Linux 8.04.
- FreeBSD Foundation выделил на 2008 год $80000, для финансирования проектов по развитию ОС.
- Руководство ISO и IEC рекомендует отклонить апелляции от Бразилии, Индии, ЮАР и Венесуэлы в отношении сертификации OOXML.

Россия присоединилась к договору ВОИС

23 июля 2008 года пресс-служба Правительства России сообщила о том, что Председатель Правительства В.В. Путин подписал Распоряжение «О присоединении Российской Федерации к Договору Всемирной организации интеллектуальной собственности по авторскому праву (ВОИС), принятому Дипломатической конференцией по некоторым вопросам авторского права и смежных прав в г. Женеве 20 декабря 1996 года».

Распоряжение было подписано премьер-министром в соответствии с пунктом 1 статьи 21 федерального закона «О международных договорах Российской Федерации». Инициатором присоединения России к договору Всемирной организации интеллектуальной собственности по авторскому праву выступило Министерство культуры. Решение о принятии договора было одобрено Министерством иностранных дел, Минобрнауки и Минкомсвязью Российской федерации.

Напомним, что ВОИС является специализированным учреждением Организации объединенных наций, которое занимается развитием международной системы защиты интеллектуальной собственности. Договор по авторскому праву описывает механизмы защиты различных видов интеллектуальной собственности, таких как литературные произведения, фонограммы и компьютерные программы.

Итак, похоже, правительство собирается взяться за пиратов всерьез. На первый взгляд, тенденция хорошая – ведь нелегальное распространение коммерческих программ очень сильно мешает продвижению открытых продуктов. Однако с реализацией конкретных процедур по борьбе с компьютерным пиратством дела обстоят не так гладко.

Другой важной проблемой является то, что юридический статус открытых лицензий (GPL, BSD и других) в Российской Федерации пока до конца не ясен. Поскольку большинство пользователей открытого ПО не имеют бумаг, подтверждающих легальность установленных на компьютере программ, у них могут возникнуть определенные трудности с контролирующими органами. Разумеется, для того, чтобы привлечь гражданина к уголовной ответственности, факт использования пиратских программ придется доказывать в суде, а в случае со свободными продуктами сделать это будет затруднительно. Однако если ваш компьютер или сервер изымут на неопределенный срок для проведения экспертизы, вряд ли это поспособствует распространению Open Source [мы подробно рассматривали эти и другие проблемы в **LXF108**, – *прим. ред.*].

В завершение мы хотим пожелать правительству успехов в нелегком деле борьбы с компьютерными пиратами. Надеемся только, что лекарство не окажется страшнее болезни, и все изменения в законодательстве, а также механизмы их реализации будут тщательно продуманы.

Что было: Linux World Expo 2008

С 4 по 7 августа в центре Moscone, Сан-Фрациско, Калифорния проходила ежегодная выставка-конференция LinuxWorld Expo 2008. В ней приняли участие около двухсот компаний, а также тысячи разработчиков и энтузиастов открытого ПО со всего мира. Конференция, как было отмечено выше, продолжалась с 4 по 7 августа, а выставочная часть Linux World Expo – с 5 по 7. За четыре дня мероприятие посетило более десяти тысяч человек.

Многие крупные корпорации анонсировали на выставке свои решения. Посетителям были доступны стенды таких компаний, как Access, Bivio Networks, Cisco Systems, Novell, Oracle, Rackable Systems, SugarCRM, Fujitsu, Intel, Talend, Texas Instruments, Ubucon, VMware и Wind River.

С речами выступили официальные лица: вице-президент и генеральный менеджер Cisco Systems Раджив Рамасвами [Rajiv Ramaswami], технический директор Citrix Systems Саймон Кросби [Simon Crosby], вице-президент IBM по открытым разработкам и стандартам Боб Сатор [Bob Sutor], главный архитектор корпорации Oracle Эдвард Скревен [Edward Screven] и многие другие.

На конференции обсуждались направления развития открытого ПО, перспективные проекты и насущные проблемы. Выставочные стенды привлекали внимание посетителей множеством интересных новинок. В частности, были продемонстрированы третья версия операционной системы gOS на основе ядра Linux и коммуникатор Openmoko Neo Freerunner.

Вообще, выступлений по поводу использования Linux на мобильных устройствах было великое множество. На фоне оптимистичных прогнозов анонсировались коммуникаторы, а в альянс LiMo вступали новые участники. Однако главный конкурент Openmoko, компания Google, не присоединилась к LiMo, зато пообещала выпустить в конце этого года собственный коммуникатор с нашумевшим Android.

Представители Hewlett-Packard заявили о планируемой в Serviceguard поддержке Novell SUSE Enterprise Linux и Red Hat Enterprise Linux и анонсировали возможность развертывания служб каталогов на основе OpenLDAP.

Alfresco и Canonical сообщили о готовящемся запуске совместного проекта по распространению недавно вышедшей бета-версии ECM-системы Alfresco Labs 3 через партнерский репозиторий Ubuntu.

Как можно видеть, прошедшая в Сан-Франциско выставка была богата интересными событиями, однако стоит заметить, что неожиданностей на Linux World Expo 2008 не случилось – обо всех серьезных анонсах и новинках было известно заранее. Впрочем, на подобных мероприятиях сюрпризов обычно не бывает. Зато есть хорошая возможность увидеть все интересное сразу в одном месте. LXF

» **Читайте репортаж с LinuxWorld 2008 в одном из следующих номеров LXF!**

LINUX FORMAT

Хронология

События в мире Open Source и около него, произошедшие в период с 30 июля по 16 августа 2008 года

» Дистрибутивы

16.08.2008 г. Проекту Debian исполнилось 15 лет. В этот день Ян Мердок [Ian Murdock] объявил о создании нового дистрибутива. Сейчас пользователям доступен Debian 4.0 (Etch), а в сентябре ожидается выход следующего релиза – Debian Lenny.

15.08.2008 г. Создатели китайского дистрибутива RedFlag Linux приурочили выход седьмой версии к началу Олимпийских игр 2008 г. в Пекине. Основные компоненты релиза: ядро 2.6.25, *glibc 2.8*, *GCC 4.3.1*, *xorg-server 1.4.99*, *Qt 4.4.0*, KDE 4.1, *Firefox 3.0*, *Wine 1.0*.

13.08.2008 г. В Slackware Current появился *Qt 4*. Началось тестирование KDE 4.1.

08.08.2008 г. Прекратил существование коммерческий дистрибутив Linspire, ранее известный как Lindows. Все разработки Linspire будут перенесены в Xandros.

06.08.2008 г. Выпущен ReactOS 0.3.6.

23.07.2008 г. ALT Linux 4.0 Desktop Professional сертифицирован ФСТЭК для работы с конфиденциальной информацией и персональными данными по 4 классу НДВ и по 5 классу СВТ.

29.07.2008 г. Вышел развивающий дистрибутив для детского творчества ALT Linux 4.0 Children.

29.07.2008 г. Пользователям стало доступно седьмое обновление Red Hat Enterprise Linux 4.

» Уязвимости

31.07.2008 г. Ицик Котлер [Itzik Kotler], менеджер по безопасности Radware, создал для конференции Black Hat троянскую программу *Jinx*, поражающую браузер *Firefox 3*.

Уязвимости подвержены все версии до релиза 3.0.1. Троян, написанный на JavaScript, индексирует и отсылает файлы с зараженного компьютера. Он работает в *Firefox 3* для Windows, Linux или MacOS X.

31.07.2008 г. Обнаружены многочисленные уязвимости в Python. Всем пользователям версий 2.4.X и 2.5.X рекомендуется обновиться.

09.07.2008 г. Дэн Камински [Dan Kaminsky] сообщил о критической уязвимости большинстве DNS-серверов. Кроме BIND, проблема затронула Cisco IOS, Juniper JunOS и Microsoft Windows DNS Server.

» Новинки открытого ПО

14.08.2008 г. Вышел Smart Package Manager 1.0. Данный менеджер не зависит от типа пакетов, применяемых дистрибутивом. В настоящее время поддерживаются форматы DEB, RPM и Slackware TGZ.

09.08.2008 г. После пятнадцати месяцев разработки выпущен *Vim 7.2*. Исправлены многочисленные ошибки и добавлена поддержка операций с плавающей точкой. Теперь скриптовый язык Vim можно использовать для расчетов.

09.08.2008 г. KDE 4.1 перенесен на FreeBSD и доступен пользователям этой системы через коллекцию портов.

07.08.2008 г. Состоялся релиз PHP 4.4.9. На этом официальная поддержка PHP4 заканчивается – обновления и патчи к четвертой версии больше выпускать не будут.

07.08.2008 г. Пользователям доступен LXDE (Lightweight X11 Desktop Environment) – «легкое» окружение рабочего стола, предназначенное в основном для ультрапортативных ноутбуков. Этот проект обратил на себя внимание после выхода gOS.

06.08.2008 г. Для тестирования доступны порты *Mozilla Firefox* на DirectFB и *Qt 4*. Предыдущие попытки портирования на *Qt* успехом не увенчались – код худо-бедно работал, но не получил официального статуса.

30.07.2008 г. QNX Software Systems продолжает открывать исходные тексты RTOS QNX. Опубликован код файловых систем. Тексты доступны на условиях QNX Non-Commercial End User License.

IBM®

ВЫСОКОПРОИЗВОДИТЕЛЬНЫЙ ЦЕНТР ОБРАБОТКИ ДАННЫХ. ЭНЕРГОСБЕРЕГАЮЩИЙ. РЕНТАБЕЛЬНЫЙ. ЭКОЛОГИЧНЫЙ.

Решения IBM CoolBlue на базе передовой технологии POWER6 позволяют снизить энергопотребление и энергозатраты без ущерба производительности. Дополнив серверы и системы хранения IBM функциями виртуализации, вы сможете оптимизировать работу тех серверов, которые используются недостаточно эффективно, и сократить очередь задач, что будет способствовать повышению экологической безопасности.

Подробности на **ibm.com/systems/green/ru**

IBM, ibm.com, логотип IBM, CoolBlue и POWER6 являются товарными знаками или зарегистрированными товарными знаками International Business Machines Corporation в США и/или других странах. Наименования других компаний, продуктов и услуг могут быть товарными знаками или знаками обслуживания третьих лиц. © 2008 IBM Corporation. Все права защищены.

LXF Репортаж

Пятая Протва

Роман Комков делится своими впечатлениями от последней конференции разработчиков свободного ПО.

Уже пятый год в Обнинске, что в Калужской области, проходит Конференция разработчиков свободного программного обеспечения на Протве, названная по реке, протекающей рядом с Обнинском и впадающей в Оку. Организует ее российская компания ALT Linux, а принимающей стороной, на чьей территории проходит мероприятие, выступает Государственный Центральный Институт Повышения Квалификации

Конференция пользуется большим вниманием со стороны не только разработчиков, но и представителей коммерческих компаний. Это неслучайно, ведь многие идеи, озвученные в докладах, будут полезны в бизнесе, а поговорить с их авторами можно прямо на месте, за кофе-брейком или во время обеда. Вокруг всего мероприятия царит непринужденная дружеская атмосфера, и это тоже неслучайно, поскольку многие участники давно знакомы между собой. Вечером можно погулять по Обнинску – это очень зеленый город.

В этом году в конференции приняли участие (в качестве слушателей) более 100 человек, было зачитано 34 доклада. Часть из них была посвящена внедрениям свободного ПО в государственные и коммерческие структуры. Выступления по этой теме сделали Анатолий Якушин и Вартан Хачатуров. Особо стоит отметить доклад Егора Гребнева «Государственные НИОКР в области ПО с открытым кодом в Евросоюзе». Основная же масса докладов была обращена к вопросам разработки свободного ПО.

Некоторые люди выступают с докладами далеко не в первый раз. Среди таких – Александр Боковой, рассказавший о *Samba*, а также Станислав Иевлев и Михаил Якшин (доклад о свободной платформе тестирования и мониторинга аппаратного обеспечения Inquisitor). Традиционно были зачитаны доклады, касающиеся дистрибутива ALT Linux. Большой интерес вызвал доклад гостя из Варшавы, сотрудника Mozilla Corporation, Збигнева Бранецки [Zbigniew Braniecki]. Его выступление называлось «Mozilla vision of the Internet world» [Взгляд Mozilla на мир Интернета]. Нельзя не упомянуть о том, что в рамках конференции был проведен круглый стол на тему «СПО в школе», который вызвал оживленную и интересную дискуссию – ведь этот вопрос сейчас особо актуален для Open Source-сообщества.

А вот что думает о мероприятии руководитель маркетинговых проектов компании ALT Linux Илья Машкин.

LXF: **Илья, вы представляете компанию-организатора конференции – ALT Linux. Как оцениваете уровень организации – справились?**
ИМ: По отзывам слушателей, докладчиков и спонсоров, конференция прошла удачно: отмечают и актуальность поднятых тем, и динамичность некоторых докладов, и жаркие любопытные дискуссии, и даже дополнительные кофе-паузы по утрам.

LXF: **Уже в пятый раз Обнинск. Ближе к Москве не планирует переселиться?**
ИМ: Проводить конференции в Москве не так... романтично, что ли. В области атмосфера более расслабленная, можно отвлечься от московской суеты, да и Обнинск – замечательный тихий город. Поэтому мы и проводим летнюю конференцию для разработчиков здесь, а зимнюю («СПО в высшей школе») – в Переяславле-Залесском.

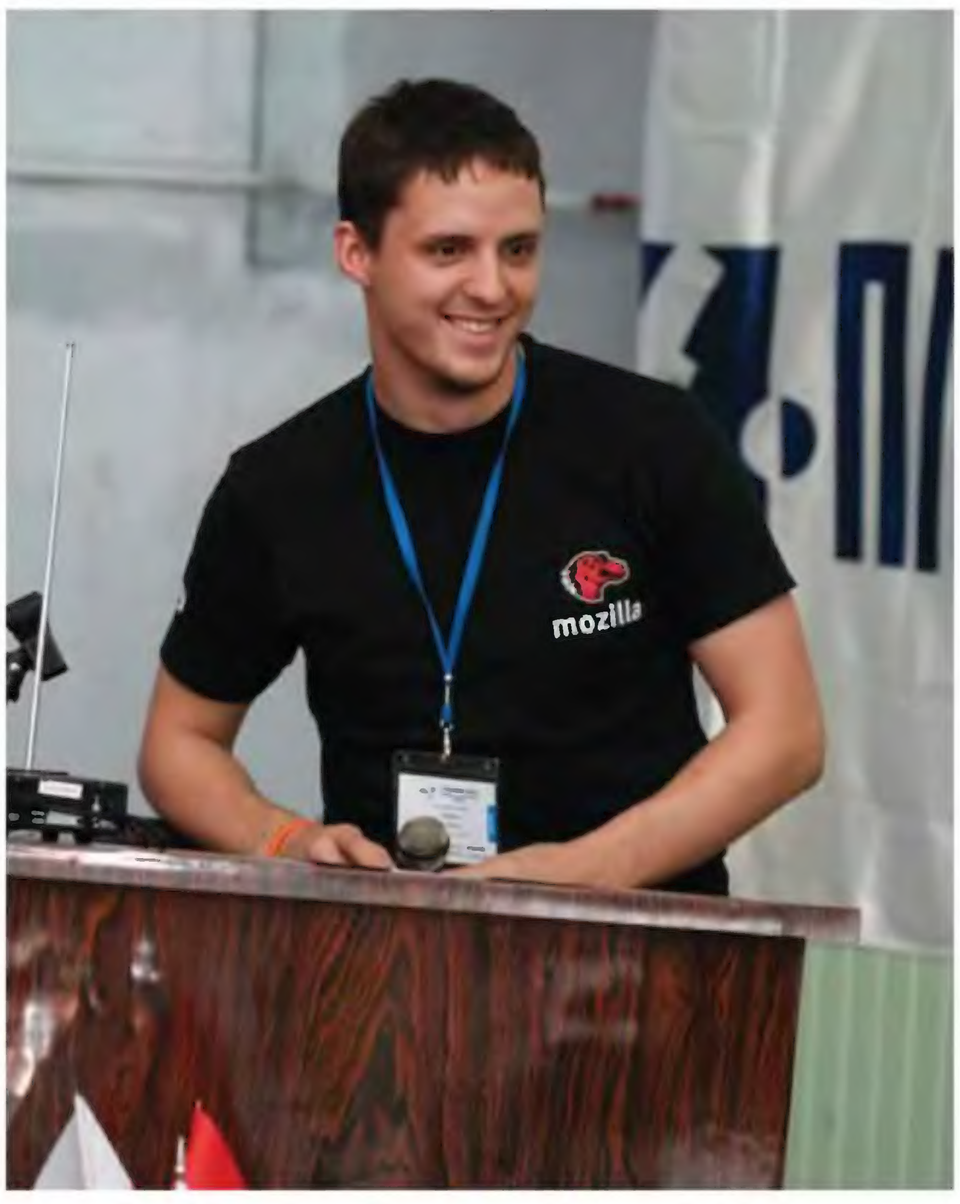


Фото: Анна Шадеева

› **Представитель Mozilla Europe, Збигнев Бранецки.**

LXF: **С каждым разом конференция становится все насыщенней и привлекает все больше и больше людей. Не думали над идеей проведения подобных мероприятий дважды в год?**
ИМ: Думали, именно поэтому, совместно с коллегами из Томской ЛУГ и Unigine, провели в мае конференцию «Свободное программное обеспечение: разработка и внедрение» (freesoft-conf.tomsk.ru) в городе Томске. Она имела большой успех и собрала порядка 350 слушателей и 40 докладчиков. В течение этого года планируем провести подобные мероприятия и в других регионах.

Подводя итоги, можно сказать, что с каждым годом конференция пользуется все большим и большим интересом со стороны разработчиков свободного ПО и людей, заинтересованных в его практическом применении. Пятая конференция закончилась совсем недавно, а мы уже с нетерпением ждем шестую. LXF

LINUX FORMAT Обзоры

Новинки программного и аппаратного обеспечения в описании наших экспертов

Алексей Федорчук
Его слабости – mass storage, разметка диска и файловые системы.

Юбилей в троичном счислении

В сентябре журналу LinuxFormat – единственному в России специализированному периодическому изданию по одноименной тематике – исполняется три года. Казалось бы, в этой дате нет ничего особенного. Однако если вдуматься, то это ничто иное, как юбилей в троичной системе счисления. А ведь реальный мир далеко не бинарен, как к тому привыкли компьютерщики: в предыдущей колонке мы видели, что даже решение о выборе дистрибутива не может быть принято на основе двоичной логики. Да и в компьютерном мире идея троичности нашла воплощение не только в теоретических построениях, но и в практически работавших машинах – достаточно упомянуть некогда знаменитую «Сетунь». Интересующимся подробностями – прямой путь в русскоязычную Википедию, там можно найти немало материалов на эту тему и еще больше внешних ссылок.

Что же произошло с нашим любимым журналом за трехлетний период его жизни? Вспомним первый (по суммарному счет оригинального издания – 70-й) его номер: целиком переводной и, не смотря на множество интересных материалов, весьма далекий от российских реалий.

А теперь откроем любой номер за текущий, 2008, год: минимум половина материалов принадлежат отечественным авторам, в том числе – законченные и продолжающиеся циклы статей по различным вопросам программирования, компьютерной графике, издательским системам. Циклы, вполне способные трансформироваться в книги – и примеры этому уже есть. Чего бы еще хотелось – это в ознаменование трехлетнего юбилея увидеть в журнале статью о троичном компьютинге – ведь идеи этого развивались в то время, когда всё программное обеспечение представляло собой Open Source, если не по букве, то по духу.

alv@posix.ru

Сегодня мы рассматриваем...

10 Black Plague

Продолжение Penumbra: Overture переносит вас в коридоры конференции Zombie Live 2008. Здесь поможет только винчестер... И не 80-гигабайтный, а 80-зарядный!

12 Qnap TS-409

4 ТБ места на четырех жестких дисках под управлением встраиваемой ОС Linux. Превосходно для коллекционеров случайных чисел и представителей малого бизнеса.

13 JBuilder 2008

Более не независимая среда для разработки на Java, JBuilder взгромоздился на плечи своего давнего соперника – открытого и свободного Eclipse. Попробуем разобраться, за что тогда берут столько денег?

14 CrossOver 7

Продолжая тему CrossOver Games прошлого месяца, Джереми Уайт и его команда наконец-то анонсировали совместимость с Microsoft Office 2007. Справится ли с ней наш бесстрашный обозреватель?

15 Blender 2.46

Если вы когда-нибудь мечтали о карьере парикмахера, этот релиз – для вас. Попробуйте новый стиль, не рискуя своими собственными волосами.

Black Plague с. 10

› Майк всегда рад ответить на звонок читателя, который забыл свой пароль.

Qnap TS-409 с. 12

› Терабайты дискового пространства и встроенный RAID делают Qnap превосходным хранилищем для ваших данных.

НАШ ВЕРДИКТ: пояснение

Все попавшие в обзор продукты оцениваются по одиннадцатибальной шкале (10 – высшая оценка, 0 – низшая). Как правило, мы оцениваем функциональность, производительность, простоту использования и цену, а для бесплатных программ учитывается документация. Кроме того, мы всегда выставляем общую оценку, демонстрирующую наше отношение к продукту.

Выдающиеся решения могут получить престижную награду «Top Stuff». Номинантами становятся лучшие из лучших – просто высокой оценки здесь недостаточно.

Рассматривая свободное ПО, мы обычно указываем предпочтительный дистрибутив. Иногда это означает компиляцию из исходных текстов, но, если разработчики рекомендуют Autopackage, мы следуем этому совету.

LINUX FORMAT Вердикт

Google Earth

Разработчик: Google
Сайт: http://earth.google.com
Цена: Бесплатно по закрытой лицензии

Функциональность	10/10
Производительность	9/10
Простота использования	9/10
Документация	9/10

» Если весь мир – сцена, то Google Earth – театр. Простая в использовании, захватывающая и ободряюще практичная программа.

Рейтинг 9/10

Penumbra: Black Plague

Мрачные коридоры? Стены в пятнах крови? Зомби-полулюди?
Это **Ник Вейч** на денек покинул Башни LXF, чтобы поискать 3D-приключений.

Вкратце...

» Трехмерная «игра ужасов». См. также: Penumbra: Overture (LXF98).

СИСТЕМНЫЕ ТРЕБОВАНИЯ:

» Процессор с частотой 1 ГГц
» 512 МБ ОЗУ
» 3D-акселератор (GeForce 3 или лучше)
» 600 МБ места на диске

Выключите свет. Закройте двери. Заприте их, если можно. Задёрните занавески – играя в эту игру, лучше быть уверенным, что за вами не подглядывают.

Black Plague [Черная чума] – продолжение предыдущей версии *Penumbra*, *Overture* (см. **LXF98**). Впрочем, не имеет особого значения, знакомы ли вы с прежней игрой и знаете ли её сюжет. Главное – помнить, что надо выбраться из гиблого места, населённого ужасными зомби-нелюдями. Впрочем, довольно реально, что ваш персонаж обернётся одним из них.

Приключения развиваются в анфиладе комнат, в основном жутких и мрачных; в каждой из них необходимо разгадать головоломку, чтобы двигаться дальше. Обычно помогают подручные предметы – например, взгромоздите друг на друга несколько ящиков, чтобы перелезть через изгородь, или сбейте замок, подобрав кирпич.

› Предлагаемые головоломки в основном логические, но требуют момента озарения.

Управление

Сами головоломки, как правило, вполне поддаются логике, но вот процесс управления весьма любопытен. Как и в *Overture*, действие в *Black Plague* происходит в реалистичном объёмном мире. Крошечное пятнышко на экране указывает, куда смотрит ваш персонаж, а его вид меняется в зависимости от предстоящего действия. Если оно превратилось в изображение руки, можно поднять или перетащить объект, подбросить его в воздух или распорядиться им как-либо иначе. На двери и другие предметы «рука» воздействует как ей и полагается.

Весомо ли чувство страха? Это зависит от настроения и от качества видеокарты. В этой игре столько всего доставляется графическими эффектами, что не обойтись без мощного оборудования. Подойдёт GeForce 6800 или лучше, причём с проприетарными драйверами (тсс!). Умелое использование эффектов размывания и сглаживания, игра света постоянно держат вас в напряжении. Графика здесь играет важнейшую роль, и если машина у вас слабовата, вы многое теряете. Да и действовать будет труднее: снижение чёткости деталей зачастую делает карты и знаки неразборчивыми, а предметы, имеющие решающее значение, сливаются с фоном.

Физический движок – не единственное средство, используемое для поддержания тонуса. Саундтрек игры способен в пять минут

› Всяческие терминалы и компьютеры, встречающиеся в игре, органично вписываются в сюжет.

нервировать кого угодно. А выключать его не стоит, ведь он часто предупреждает о приближении опасности. Ну, а закадровый голос – это нечто!

Но правдоподобное моделирование реального мира имеет свои пределы. Чрезмерное количество текстур замедляет игру до подвисания; в комнате, битком набитой даже простейшими предметами, становится трудно что-то отыскать. Физическая модель взаимодействия с объектами тоже раздражает. Хотя это реалистично и неизбито – открыть ящик, «взявшись» за него и «потянув» на себя мышью, иногда даже кликнуть по рукоятке бывает не так-то просто. Бесконечные ряды ящиков, которые предстоит открыть, наводят ужас почище «заражённых» (игровое прозвище зомби-гуманоидов). Вообще-то ужасные создания встречаются не так часто, и ваш персонаж имеет больше шансов неудачно упасть, утонуть или угодить в случайную ловушку. Скорее всего, вы будете не убиты, а просто застрянете в тупике.

К счастью, головоломки по большей части вполне логичны. Хотите получить стакан воды из автомата? Для этого есть как минимум два способа. Тем труднее разобраться в ситуациях с элементами реальной сложности. По большей части необходимые вещи находятся в пределах видимости, но иногда можно растеряться, не разобрав, что нарост на потолке на самом деле – выключатель, до которого нужно дотянуться.

В некоторых местах игра кажется однообразной, так как препятствия, с которыми вы встречаетесь, вам вроде бы уже попадались, но обычно она оставляет чувство удовлетворения от выполнения особо сложной миссии. Точно зная, что и как делать, игру можно пройти за четыре-пять часов, но вы сами не захотите с ней расстаться. LXF

› Световые эффекты работают на гнетущую атмосферу. И – не верьте надписям на стенах...

› Один из самых жутких моментов игры, после которых особенно остро хочется на свежий воздух.

Реальный мир?

Физический движок и игровая логика рассчитаны на придание игровому миру максимума реализма. Предметы можно поднимать и поворачивать, можно швырнуть через комнату или перетащить по полу, если уж очень тяжёло поднять. Игровая логика усиливает физические эффекты. Например, чтобы сдвинуть рычаг, можно по-простому «взять» его виртуальной рукой и повернуть. А можно, смеха ради, подобрать ящик или кирпич и уронить его на рычаг – тоже сработает. Хотите – пните его, хотите – постройте адскую машину из приводов и противовесов, отойдите в другой угол комнаты и бросьте на рычаг камешек – система не подведёт. В этом смысле игра весьма реалистична, что придаёт ей особый колорит.

› Виртуальная рука только одна! И как тут достать кетчуп из бутылки?

LINUX FORMAT Вердикт

Penumbra: Black Plague

Разработчик: Frictional Games
Сайт: www.paradoxplaza.com/penumbra
Цена: 19,99 фунт. ст.

Сюжет	7/10
Графика	9/10
Увлекательность	7/10
Оправданность цены	9/10

» *Превосходит предшественницу,* Overture. *Очень не похожа на традиционные Linux-игры.*

Рейтинг 8/10

Qnap TS-409 Turbo

Грэм Моррисон разыскал 4-терабайтный сетевой накопитель для своей растущей коллекции музыкальных файлов.

Вкратце...

» 4-терабайтный RAID-накопитель со встроенным Linux. Среди недорогих альтернатив – Excito Bubba, Qnap 109 Pro или ваш собственный, специально настроенный ПК с Linux.

Серверы

» Samba
» NFS
» AppleTalk
» SSH
» BitTorrent
» FTP
» iTunes
» UPnP
» MySQL
» SQLite

Мы оценивали парочку сетевых накопителей совсем недавно. В **LXF104** малыш Bubba получил наш заветный "Top Stuff", в том числе и за предоставление полного доступа к ОС Debian. Ещё раньше, в **LXF99**, мы рассматривали Qnap TS-109 Pro.

Оба устройства основаны на встраиваемых версиях Linux, обладают высокопроизводительными жёсткими дисками и могут служить заменой ПК, когда необходим накопитель с сетевым доступом и низким энергопотреблением. Qnap TS-409 более амбициозен. Вместо скрытой ячейки единственного винчестера здесь четыре отдельных гнезда, легко доступных при снятой передней решётке. В каждое можно вставить диск SATA I/II объемом до 1 ТБ, с возможностью «горячей замены». Мы даже попробовали удалить один из дисков во время работы устройства, чтобы посмотреть, что случится. Раздался предупредительный сигнал, светодиодный индикатор загорелся красным, но потери данных не произошло. Благодаря избыточности RAID 5 недостающие данные одного диска были восстановлены с оставшихся трёх. Мы вставили диск на место, и через несколько мгновений зелёный цвет индикатора сигнализировал о восстановлении целостности массива.

Внешне TS-409 больше напоминает сабвуфер, чем NAS, и весит примерно столько же (много!). На передней панели – USB-порт для внешнего накопителя, кнопка моментального копирования с внешнего источника и светодиодные индикаторы активности для каждого диска. На задней стенке – ещё два USB-порта, разъём электропитания и, естественно, гнездо гигабитного Ethernet. Охлаждением занимается всего один кулер, с адаптацией скорости к нагрузке приводов, но большую часть времени он молчит. При перегрузке, сочетание шумов дисков, вибрации корпуса и работающего на высоких оборотах вентилятора делают устройство слишком шумным для размещения в гостиной, но средний ПК «жужжит» примерно так же.

› Пользователи одной сети имеют доступ к общим папкам, каталогам загрузки, web-директориям и USB-устройствам с Qnap TS-409 Turbo.

В TS-409 используется такой же ARM-процессор 500 МГц, что и в TS-19; оперативная память та же (256 МБ плюс 8 МБ флэш). К сожалению, производительность по сравнению с прежней версией не выросла, но ресурсов для выполнения большинства серверных задач хватает (получше Excito Bubba с 200 МГц). Например, *SlimServer* на Bubba разворачивался с трудом, а на TS-409 работал без всяких проблем.

Web-интерфейс

Linux-ПО на сопроводительном диске нет, но web-интерфейс легко обнаружить, если просмотреть выданные DHCP-адреса. Здесь, с помощью несложного пятишагового мастера, можно настроить основные параметры системы. По умолчанию включается форматирование и мониторинг RAID, NFS, общий доступ по Samba и Apple Talk, загрузка по BitTorrent и медиа-трансляция UPnP через встроенный сервер *TwonkyMedia* (см. наш обзор в **LXF96**). Пользовательские квоты доступны по щелчку мыши, а доступ по SSH действует по умолчанию. С главной страницы можно активировать комбинацию *Apache/MySQL*, PHP и CMS-систему *Joomla*. Эти функции делают TS-409 идеальным устройством для корпоративной сети небольшой компании (хотя Slashdot ему не по плечу). В общем, превосходный выбор для тех, кому нужно много места и надежность RAID, и конечно же, он работает под Linux. LXF

Свойства навскидку

Горячая замена
Преднастроенный RAID позволяет удалять диски и ликвидировать неисправности без потери данных.

Web-доступ
Любой сервер можно настроить через web-интерфейс, и обращаться к командной строке не придётся.

LINUX FORMAT Вердикт

Qnap TS-409 Turbo

Разработчик: Qnap Systems
Сайт: www.qnap.com
Цена: ок. $1000 (без дисков)

Функциональность	9/10
Производительность	9/10
Простота использования	8/10
Оправданность цены	7/10

» *Довольно дорого, но мы не знаем другого устройства на Linux, обладающего подобными функциональностью и вместимостью.*

Рейтинг 9/10

JBuilder 2008

Оторвавшись от хитросплетений C++/Qt, **Грэм Моррисон** сварил кофейку и даже успел немного попрограммировать.

Вкратце...

» Интегрированная среда для разработки Java-приложений любого масштаба. Использует *Eclipse*, который по совместительству служит единственным соперником.

Поддержка серверов

» Borland ES
» BEA WebLogic
» IBM WebSphere
» Oracle AS
» OC4J
» Apache Geronimo*
» Apache Tomcat*
» JBoss*
» Glassfish V2*

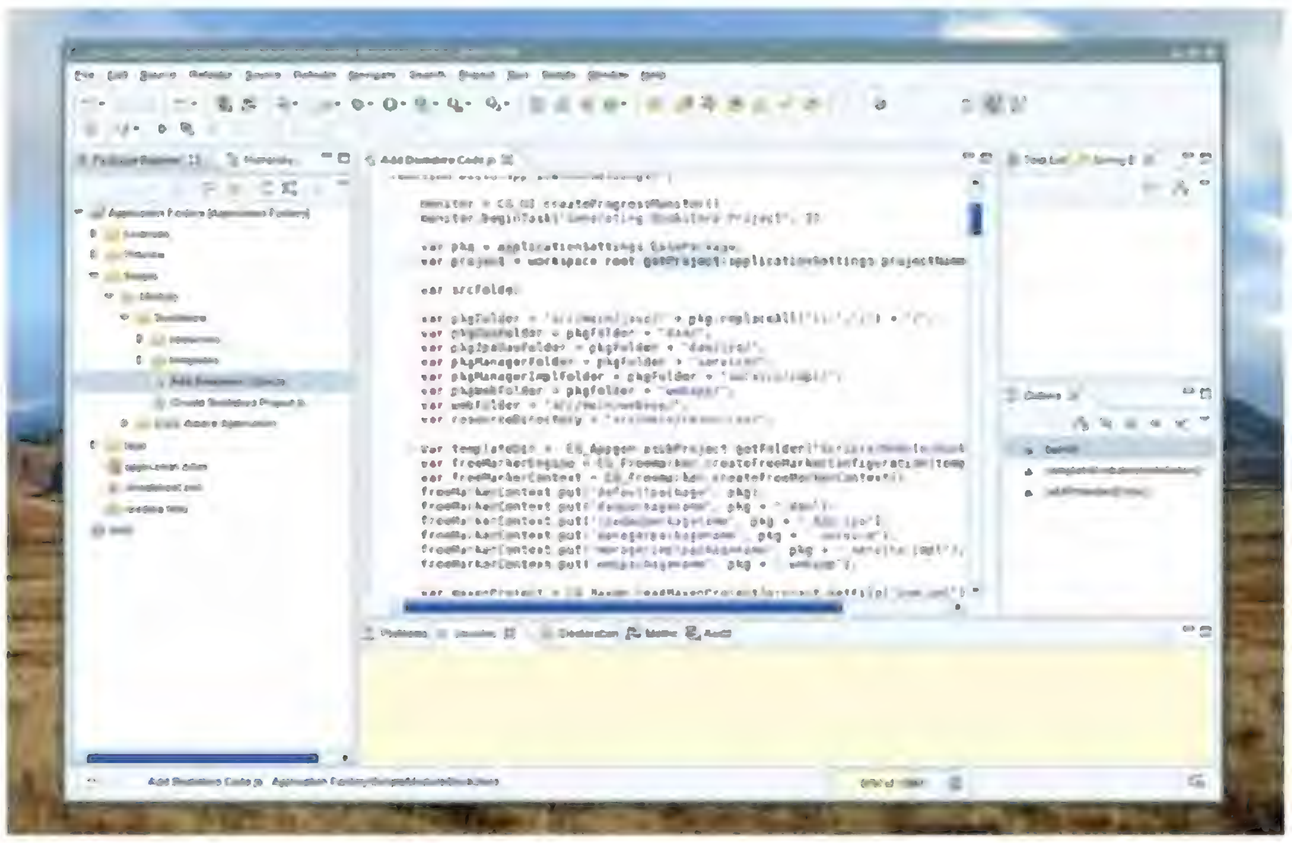
(*включено по умолчанию)

Последний раз мы рассматривали *JBuilder* в **LXF74/75**. Тогда этой превосходной, но и дорогой Java IDE хронически наступал на пятки открытый конкурент – *Eclipse*. Помнится, мы даже вставили в подзаголовок статьи астрономический каламбур [*Eclipse* по-англ. затмение]. С тех пор *Eclipse* неуклонно развивался, и в конце концов стал стандартом Java-разработки. К тому же IDE ценой в сотни фунтов с трудом вписывается в пейзаж свободного ПО.

И авторы *JBuilder* сменили стратегию: что называется, «не побил, так побратайся». Начиная с версии *JBuilder 2007*, у пакета больше нет отдельного интерфейса, и конкуренция завершилась слиянием с *Eclipse*. *JBuilder* стал модулем расширения *Eclipse* и частью *Eclipse* Workbench, усилив стандартную конфигурацию специфическими функциями.

К сожалению, с установкой *JBuilder* как на Mandriva Spring 2008, так и на новейшем Ubuntu не обошлось без затруднений. Дело в том, что основное Java-приложение напрочь отказывалось предоставлять какие-либо параметры установки. Мы решили проблему, разыскав исполняемые файлы компонентов вручную, но кому охота с этим связываться? Официальная поддержка распространяется лишь на Red Hat Enterprise Linux 4.0 – весьма неумно, если учесть, что дистрибутиву уже больше трёх с половиной лет. Помимо этой «болезни роста», никаких проблем со стабильностью мы не обнаружили.

Тех, кто обновляется с *JBuilder 2006*, ошеломит разница интерфейсов. Выверенной, лощёной рабочей среды прежней версии больше нет: получите 12 разных панелей, втиснутых в единственное окно. Раскладка далека от аскетической – даже *KDevelop* в сравнении выглядит консервативным. Да, можно позакрывать лишние окошки и передвинуть панели, да и в меню Window Perspectives (Перспективы) немало готовых вариантов для целого ряда обычных задач. Тем не менее, это усложняет и без того непростой процесс изучения пакета.



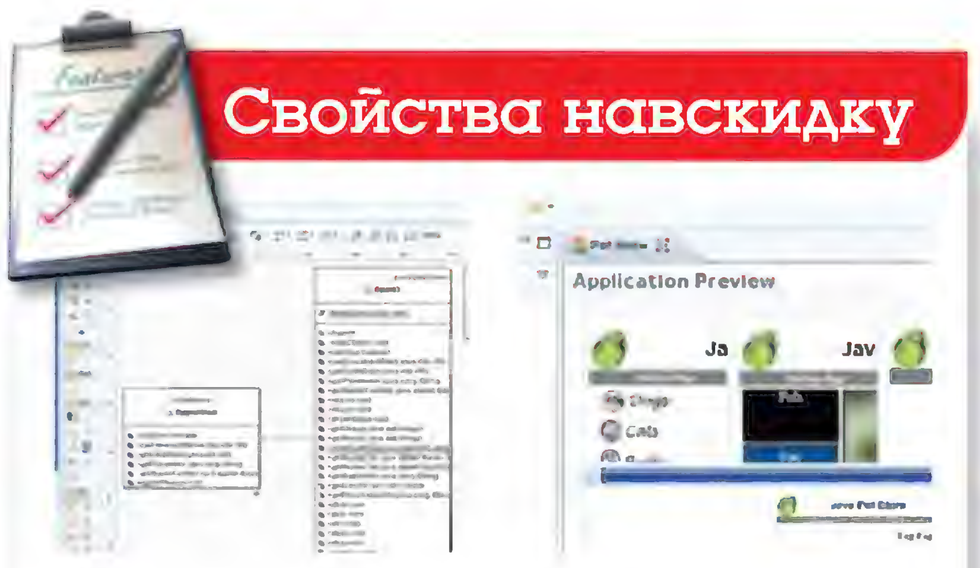
› ***JBuilder 2008*** **– второе поколение *JBuilder* на основе *Eclipse*, использующее структуру *Eclipse 3.3*. Полюбуйтесь, сколько изменений в новом интерфейсе!**

Новые возможности

Но не всё так плохо. За новыми панелями скрывается функциональность, которая вполне стоит своих денег. Наиболее заметная функция называется 'Application Factories' (Фабрики программ). Это не просто шаблоны приложений, ускоряющие процесс программирования – это централизованные хранилища проектов, над которыми программист трудится повседневно. Идея в том, что разработчики могут присовокуплять к исходным шаблонам изменения и скрипты, полученные в ходе работы над конкретными вариантами программ. «Заготовочная» версия проекта – динамическое представление типичного рабочего процесса программиста, и мы ожидаем высокую эффективность такого подхода к Java-разработке: он поможет не изобретать велосипед. Версия 2008 отмечена возвратом *Swing* в *JBuilder*, с добавлением *Swing Designer* сторонней разработки. В профессиональной версии добавляется профилирование и анализ кода, а версия Enterprise предоставляет инструменты групповой работы, включая мониторинг хранилищ и отслеживание процесса разработки.

Если вы уже работали в *Eclipse*, на освоение *JBuilder* понадобится пара дней, и нам кажется, что добавленная функциональность оправдывает свою цену. Но только для тех, кто зарабатывает Java-программированием. Хотя *JBuilder* весьма недёшев – особенно в сравнении с другими аналогами – его ценность зависит от места, которое в вашей деятельности занимает Java-разработка. Профессиональный Java-программист окупит новый инструмент с лихвой. LXF

Свойства навскидку



UML-моделирование

Интегрированное графическое UML-моделирование и дизайн объектов с синхронизацией в коде – лучшие в своём классе.

Заготовки программ

Выберите из библиотеки готовое приложение или создайте собственный шаблон для упрощения дальнейшей разработки.

LINUX FORMAT Вердикт

JBuilder 2008 Enterprise

Разработчик: CodeGear
Сайт: www.codegear.com
Цена: от 36620 руб.

Функциональность	9/10
Производительность	9/10
Простота использования	7/10
Оправданность цены	6/10

» *Достойное приобретение для программистов-профессионалов.*

Рейтинг 7/10

CrossOver Linux 7

Не это ли приложение поможет робким Windows-пользователям перейти на Linux? **Саймон Пикстон** загружает Office 2007 и пробует…

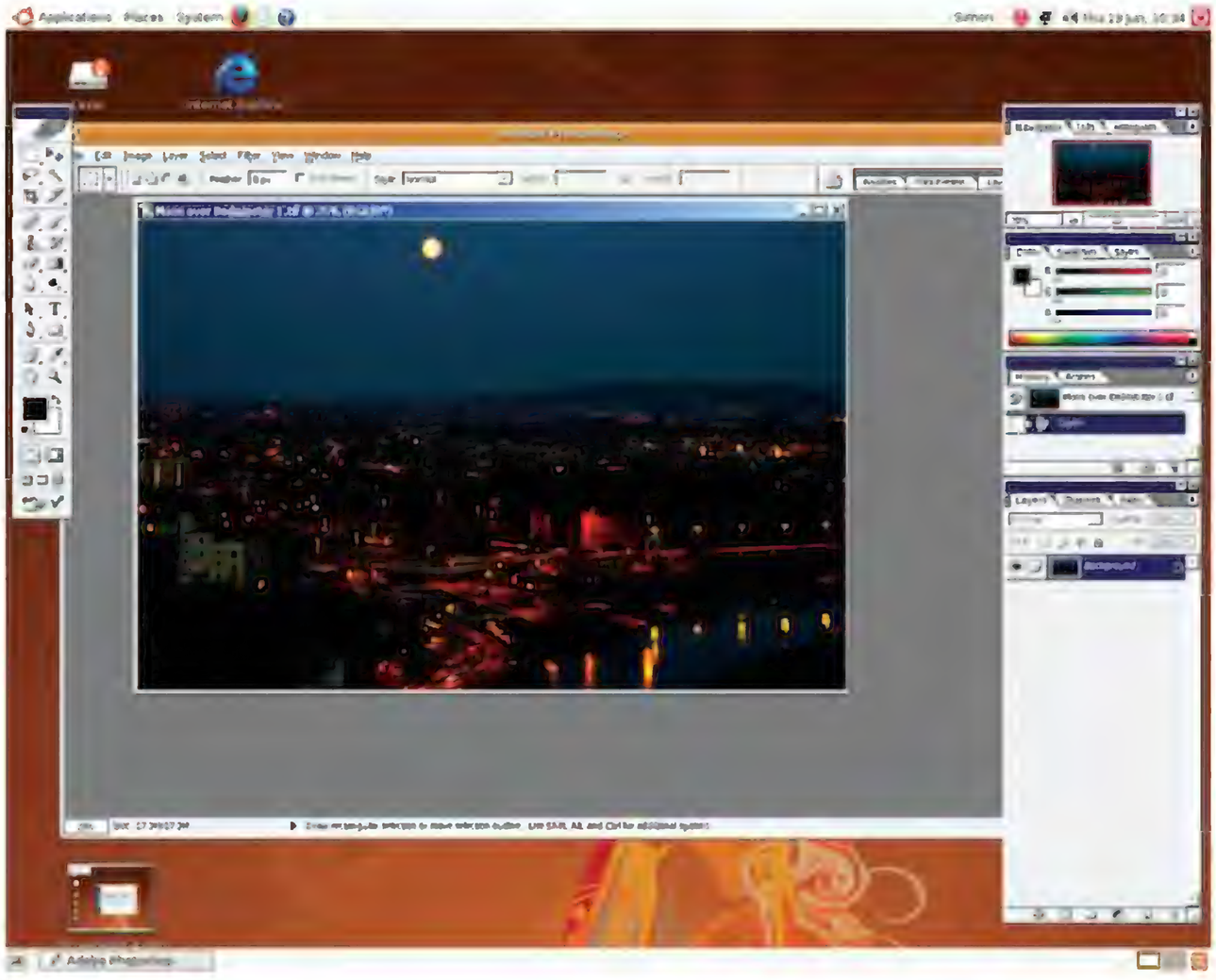
Вкратце...

» Слой совместимости, позволяющий Windows-пользователям запускать свои приложения в Linux. См. также: *Cedega*, *Wine*.

Перемен боятся все, и среднему Windows-пользователю Linux представляется опасным путешествием в неведомое – таинственным и недоступным простым смертным. Да еще распускаются слухи, что Windows-приложения не работают в Linux. У *OpenOffice.org* и *GIMP*, при всех их достоинствах, не так уж много приверженцев вне мира свободного ПО. Здесь на помощь приходит *CrossOver Linux*, позволяющий запускать *Office* и *Photoshop* вне рабочей среды Windows (на Linux и Mac OS).

Седьмая версия *CrossOver Linux* рассчитана на поддержку новых приложений, например, *Microsoft Office 2007* и некоторых современных игр. Принцип действия *CrossOver* заключается в образовании «бутылок» (оболочек, эмулирующих различные версии Windows). После загрузки и установки *CrossOver* в виде Debian-пакета, можно выбирать между инсталляцией Windows-ПО и выполнением команд Windows. Для инсталляции выводится перечень ПО, поддерживаемого *CrossOver*, а командой можно найти какой-либо setup-файл, которого в списке нет.

Первым на очереди был *Microsoft Office XP*. Хотя *OpenOffice.org* – вполне достойная альтернатива, многие компании считают, что отсутствие этикетки Microsoft на упаковке подрывает его репутацию. Офис запустился без проблем, хотя *CrossOver* почему-то решил, что ему нужна оболочка Windows 98, а не XP, как мы ожидали. Но после инсталляции все основные приложения пакета действовали безупречно.



› **Adobe *Photoshop CS* устанавливается как по маслу и дружит с Hardy Heron.**

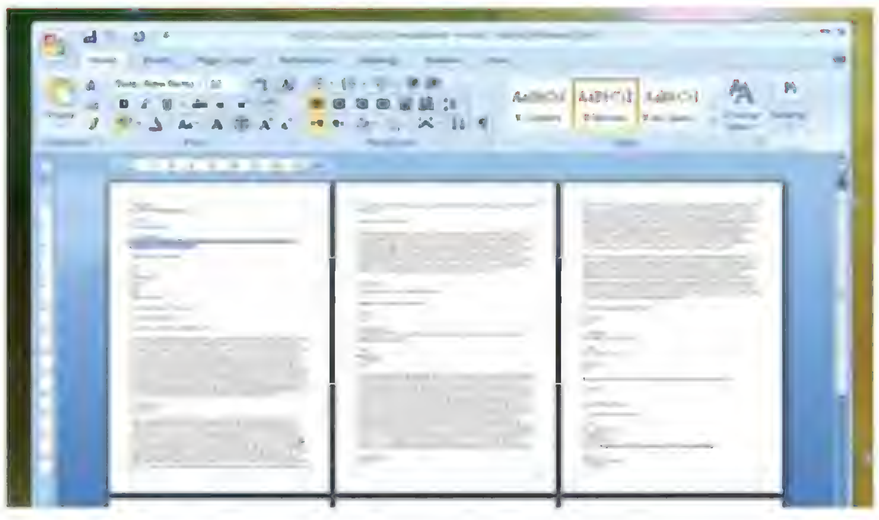
Windows-хиты

Adobe Photoshop поддерживается в обеих инкарнациях, CS и CS2. После установки *Photoshop CS* (несложной) приложение работало отлично. С *Adobe Indesign CS* было сложнее, так как его не было в списке поддерживаемого ПО, и установку «заклинило» где-то при 98%.

Приобретая Pro-версию *CrossOver*, вы получаете *CrossOver Games*, это позволяет играть во многие Windows-игры и использовать онлайн-платформу *Steam*. Правда, многие игры рассчитаны на DirectX, а не OpenGL, и список поддерживаемых брэндов пока невелик. Но это всё же лучше, чем «сидеть» на *Tux Racer*!

Настоящего линуксоида вряд ли прельстит перспектива платить за что-либо, а вот многие беженцы из Windows охотно купят *CrossOver Linux*. $40 за стандартную версию и $70 за профессиональную – невелики затраты, при этом пользователю при смене операционной системы доступно большинство привычных приложений. Да, некоторые программы не совсем устойчивы, но те, что работают, работают надёжно. LXF

MS Office 2007



› **Хотя новый интерфейс 2007 требует некоторой привычки, ясно, что *OpenOffice.org* есть чему поучиться.**

Поддержка *MS Office 2007* для этой версии – главная задача. Мы загрузили 30-дневную пробную версию с сайта MS и запустили мастер *CrossOver*. Всё установилось без запинки, нам удалось даже активировать ключ пробной версии онлайн. Через 15 минут *Word*, *Excel* и *Outlook 2007* запускались из KDE-меню. *Word* и *Excel* работают нормально, и мы настоятельно рекомендуем *CrossOver 7* всем, кому необходимы приложения *Office 2007*. Только сначала попробуйте демо-версию.

LINUX FORMAT Вердикт

CrossOver Linux 7.0.0

Разработчик: CodeWeavers
Сайт: www.codeweavers.com
Цена: $39,95 Standard, $69,95 Pro

Функциональность	8/10
Производительность	7/10
Простота использования	7/10
Оправданность цены	8/10

» *Хотя некоторые программы могут капризничать, основные приложения и устанавливаются, и работают гладко.*

Рейтинг 8/10

Blender 2.46

Дрожи, *VariCAD*; скройся, *Cinema 4D*! У **Бена Харлинга** слюнки текут при виде возможностей нового *Blender*.

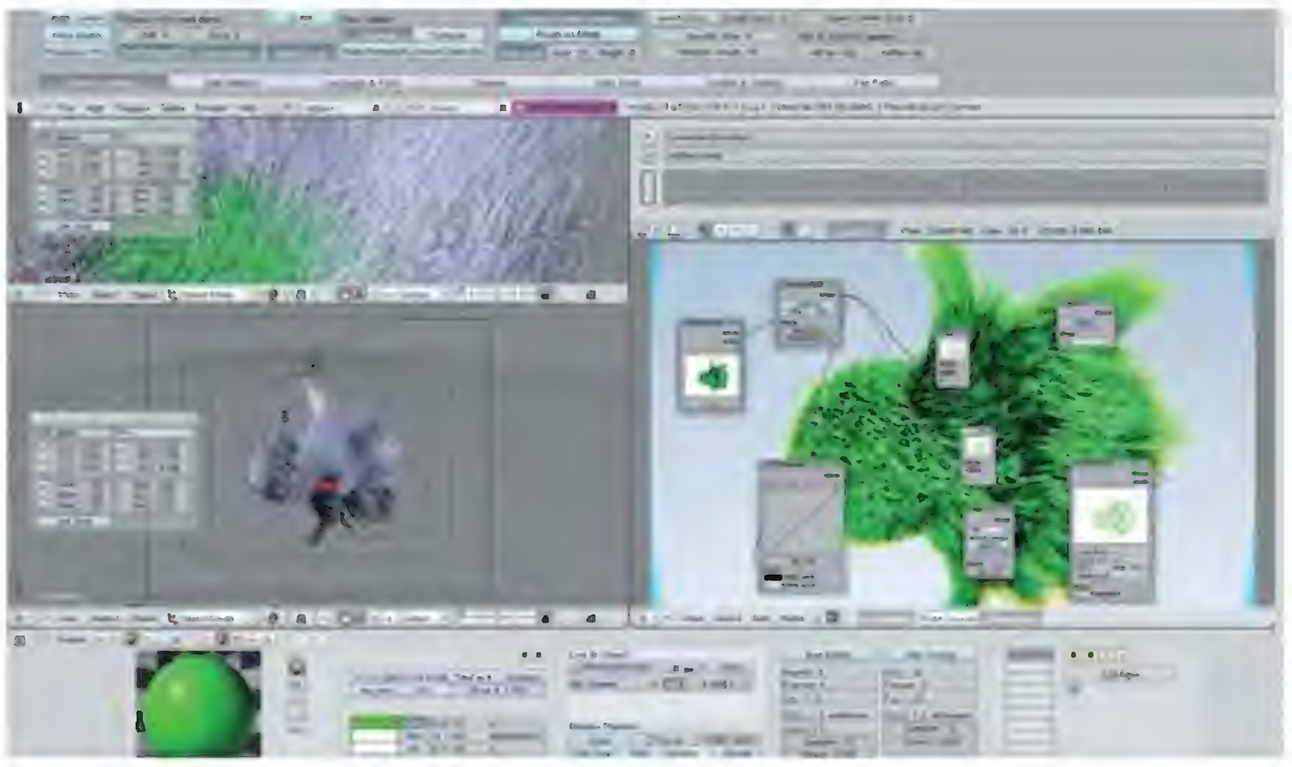
Вкратце...

» С помощью *Blender* можно производить великолепную 3D-анимацию и даже полноценно редактировать видео. См. также: *POV Ray*, *VariCAD*, *Maya*, *Cinema 4D*.

Поклонники CG-анимации, вероятно, знают, что Саша Гоедгебур [Sacha Goedegebure], режиссёр новейшего фильма *Big Buck Bunny* от Blender Foundation, был ключевой фигурой CGOverdrive, крупнейшего азиатского шоу компьютерной графики в этом году. Команда фильма испытала вкус известности, телевизионные выступления, гала-премьеру и десятки приглашений на встречи по обмену опытом. Фильм замечательный: он прекрасно реализован и по-настоящему весёлый. Но основная его задача – показать новые возможности анимационного пакета *Blender 2.46*.

А возможностей – уйма, и наш обзор упомянет только самые заметные. Первым на ум приходит небывало реалистичное отображение меховой фактуры: моделирование системы частиц воистину фантастическое. Причешите своё пушистое творение – прямо в реальном времени мех натурально заструится между зубьями расчёски. Волны дыма, взрывы, фейерверки – всё для вас, и они обновляются «на лету», мгновенно выдавая результат.

Добавлена масса «умных» инструментов для моделирования персонажей вроде тех, что участвуют в фильме *Big Buck Bunny*. Весьма вероятно, что *Blender* в один щелчок мыши завершит модель на созданном вами скелете, благодаря улучшенному механизму автоматического создания оболочки. И наоборот, собственноручно «смастерив» одежду, вы получите поразительно толковый контроль над анимацией. Любой аспект создания и визуализации каркаса поддаётся настройке, обеспечивающей широчайшие перспективы создания моделей. Наконец, функция создания оболочек типа “dual-quaternion” позволяет бесконечно совершенствовать деформацию покрытий, и превосходная анимация создается без лишней головной боли.



› **Бесконечное множество функций *Blender* неминуемо загромождает интерфейс.**

Долгое время *Blender* не хватало имитации мягких тел/материи. Данная версия это поправила: здесь надёжная система отображения и развевающейся ткани, и колышущегося студня. Работает хорошо, настроить нетрудно, но главное – тесная интеграция с основным приложением. В простых сценах можно работать с тканями и телами непосредственно, подключая ветер, турбулентность и взаимодействие с твёрдыми телами, причём в реальном времени, как и с частицами.

Да будет свет!

От всего этого, однако, мало проку без достойной отрисовки. И здорово, что встроенный движок рендеринга *Blender* был капитально отремонтирован, получив новые замечательные качества. Бывший эксплуататор процессора – перемещение источника освещения (при котором свет «пляшет» в кадре) – дело теперь вполне обычное: новый набор настроек Approximate AO обеспечивает быстрый высококачественный рендеринг.

Ещё одна новая функция – усовершенствованное отображение световых бликов, возвысившее стеклянные и металлические объекты до новой степени реализма. Менее заметны, но тоже важны новые функции сэмплинга, например ‘квази-Монте Карло’ или полноэкранное сглаживание, повышающее качество и уменьшающее количество артефактов. Полезная функция для создателей компьютерных игр – способность вносить обработанные данные рендеринга в текстуры для использования в реальном времени. Могут обрабатываться данные полного рендеринга, перемещения источника света, обычного отображения и нескольких других типов. Обработка происходит в соответствии с UV-координатами.

В свободное от создания 3D-изображений для *Linux Format* время я работаю в медиа-студии, где CG применяется сплошь и рядом. В доказательство выдающихся способностей *Blender* можно привести пример нашего последнего проекта в *Autodesk 3DS Max*. Единственной альтернативой приобретению модуля расширения за тысячи фунтов был переход на *Blender*. Клиент остался чрезвычайно доволен результатами, и есть надежда, что *Blender* прочно займёт место в рабочем процессе нашей студии, как это уже произошло во многих компаниях по всему миру. LXF

Шаг за шагом: Мех

Добавим волосы

Добавим новую систему взаимосвязанных частиц, установим параметр источника ‘hair’ (волосы), для правдоподобия внесём незначительный элемент хаоса.

Причешем шарик

В диалоговом окне нового режима системы взаимосвязанных частиц выберем параметр ‘comb’ (расчёска), и придадим нашей шевелюре импозантный вид.

Вердикт

Blender 3D 2.46

Разработчик: Blender Foundation
Сайт: www.blender3d.org
Цена: Бесплатно под GPL

Функциональность	10/10
Производительность	10/10
Простота использования	9/10
Документация	9/10

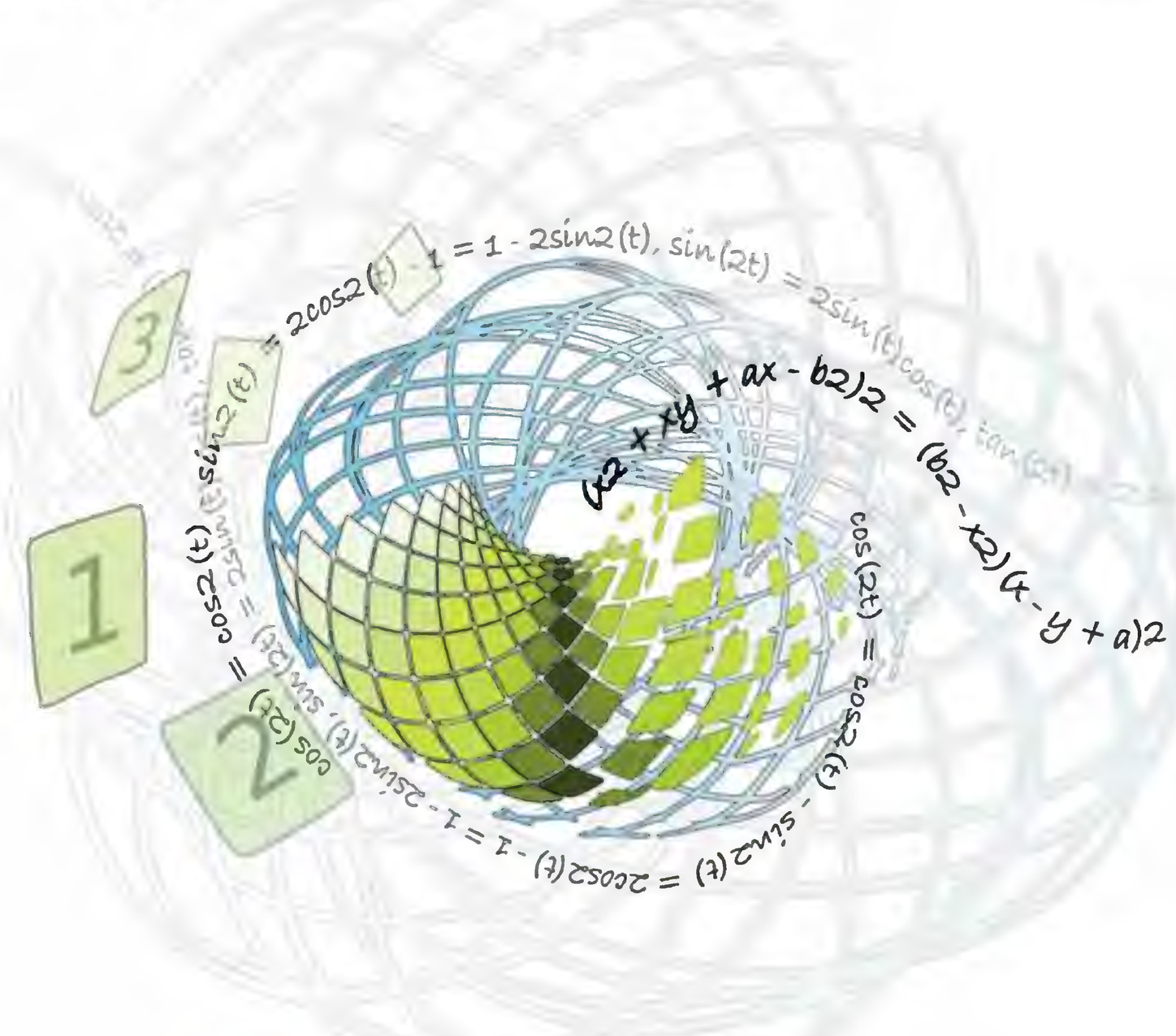
» *Трудно сказать, чего «не умеет делать» Blender. Бесподобная мощь!*

Рейтинг 10/10

» Каждый месяц мы сравниваем тысячи программ — а вы можете отдыхать!

Числодробилки

Когда ни бумага с карандашом, ни калькулятор не справляются с расчетами, **Уилл Трибби** обращается к одной из математических программ.



Про наш тест...

У платформ численных расчетов, включенных в это Сравнение, много общих функций. Значение имеет и производительность алгоритмов, что важно при работе с большими объемами данных.

Оценка выводилась по следующим факторам: простота установки, интерфейс пользователя, документация, поддержка сообщества и доступные инструменты. Производительность алгоритмов проверялась путем вычисления разложения по особым значениям (SVD) и по 100-кратному вычислению быстрого преобразования Фурье (FFT, БПФ) для матрицы действительных чисел размера 500x500, и вычисления среднего и стандартного отклонения этих ста попыток.

Тестирование проводилось на ноутбуке HP Pavilion dv5040us с процессором AMD 2,2 ГГц и ОЗУ 2ГБ, работающем под управлением Kubuntu 8.04. Все программы, кроме *Matlab* и *Euler Math Toolbox*, перед тестированием компилировались на ноутбуке.

Наш выбор

Euler Math Toolbox с. 19
Matlab с. 17
Octave с. 18
Sage с. 19
Scilab с. 20

В старые времена (1980-е) студентам ВУЗов разрешалось производить вычисления, расчеты линейной алгебры и т.п. с помощью калькуляторов – но не для всего, поскольку в основном полагались достаточными бумага и карандаш, а калькуляторы допускались лишь в особо сложных случаях. Аспиранты переходили уже к написанию собственных программ для численных расчетов, просиживая долгие часы перед терминалом за отладкой кода. Но был способ и получше.

То, что начиналось как библиотеки функций для решения численных задач, развилось в мощные, интерактивные и программируемые инструменты, «перемалывающие» сложные математические и инженерные проблемы за несколько минут. Эти платформы численных расчетов сегодня являются рабочими лошадками ученых и инженеров. Они используются практически в любой области современной науки и техники для управления данными, анализа и визуализации.

Золотой стандарт

Среди рассматриваемых здесь платформ, *Matlab* является золотым стандартом, с которым сравниваются все остальные инструменты. Это сравнение кажется неизбежным для любой платформы численных расчетов, свободной или нет. Однако, по сравнению с ними, *Matlab* имеет огромный недостаток – дороговизну. Выбор в пользу *Matlab* зависит от того, много ли дополнений вам нужно и готовы ли вы платить за них. Тем из нас, кто ограничен бюджетом, без мощной и свободной замены не обойтись.

При поиске альтернатив *Matlab*, две были уже наготове: *Scilab* и *Octave*. Обнаружение еще двух потребовало просеивания web-сайтов существующих проектов. Программы с последним обновлением старше года не рассматривались. В обзор попали наиболее обещающие и интересные решения, чьи сайты проявляют активность.

Matlab

Когда вы заслуживаете (и можете позволить) самого лучшего.

Matlab в завидном положении: все прочие лишь догоняют его, поскольку это основная вычислительная платформа, используемая большинством ученых мира и технических сообществ. *Matlab* включен в технические и математические курсы многих университетов, разносящих его репутацию также и за свои пределы. Немного поработав в среде *Matlab*, вы понимаете, почему *Matlab* устанавливает планку, до которой другим прыгать и прыгать.

Прежде всего, *Matlab* предоставляет современную IDE для выполнения вашей работы. В ней есть окно команд, окно истории команд и менеджер файлов. Команды помечаются временем и датой, и вы можете выбрать команду или группу команд в окне истории, чтобы выполнить их вновь. Вероятно, тем, кто собирается создавать лишь мелкие приложения в несколько файлов, IDE не так уж и важна. В противном случае –это большой плюс. *Matlab* часто используют для создания больших и сложных приложений, требующих управления кодом. Ни одна из рассматриваемых в обзоре платформ и близко не подошла к IDE вроде той, что есть в *Matlab*.

Матричный язык *Matlab* легок в изучении, и, освоившись с векторами и матрицами, вы сочтете его очень естественным. Несложные матричные операции, требующие нескольких операторов, или запутанные перемножения, подразумевающие целую программу, могут выполняться совместно в одной строке кода. Базовая версия *Matlab* поставляется с умопомрачительным набором математических функций, с которым остальные продукты стараются идти в ногу. Не так давно в *Matlab* было добавлено объектно-ориентированное программирование, и это также возвышает его над большинством других платформ обзора. Данный функционал может облегчить создание больших и сложных приложений для тех, кто предпочитает объектно-ориентированный подход. *Matlab* также может сразу оснастить ваши приложения графическим интерфейсом, что делает разработку более последовательной, чем при добавлении его позднее.

Книги

Для любителей пачек бумаги, испещренных черными закорючками, налицо огромный выбор. Если, зайдя в своей любимый магазин в сети, вы поищете книги, описывающие *Matlab*, вас ими просто завалят. Некоторые учебники по *Matlab* заодно описывают *Scilab* и *Octave*, но есть книги и исключительно о них. Часть книг просто дублирует документацию или учебники из сети, и иногда это удобно – особенно если вы хотите почитать в местах, где ваш ноутбук может быть в опасности. Единственная описываемая платформа, для которой мы не нашли книг – *Euler Math Toolbox*.

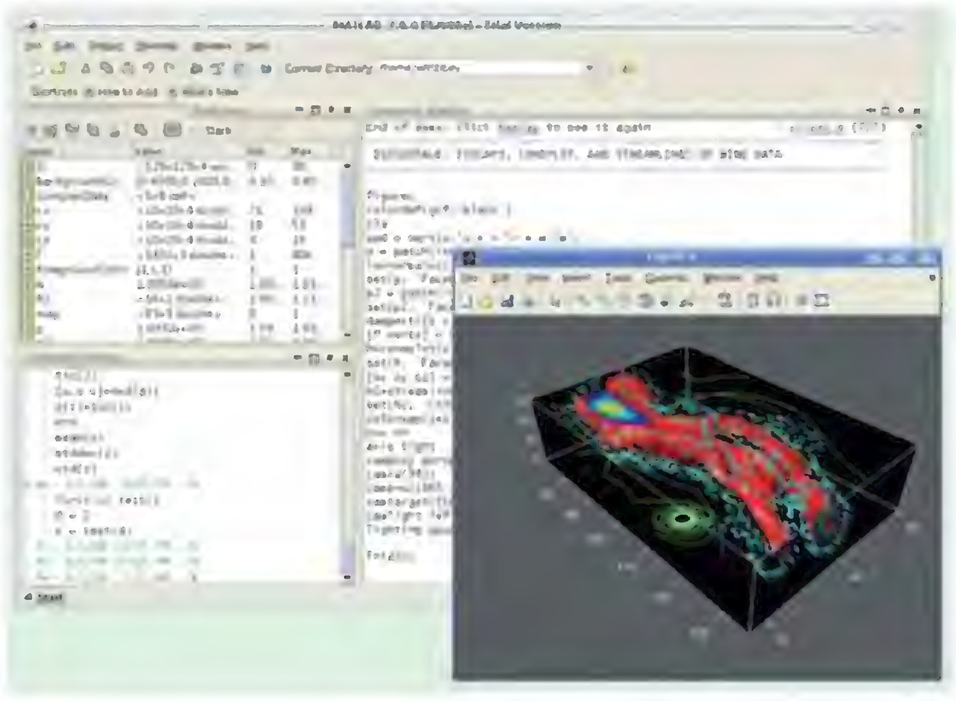
Да он еще и быстр

Производительность – та область, где *Matlab* трудно удержать лидерство. В SVD-тесте – разложении матриц, важном для многих статистических и численных алгоритмов – производительность *Matlab* лишь на 20% больше, чем в *Scilab* или *Octave*. Удивительно то, что алгоритм Numpy, используемый в *Sage*, был в среднем быстрее *Matlab* на 250%. Он не включен в настоящее сравнение, но такое разительное отличие показывает, что в библиотеке Numpy для Python производительности уделено больше внимания. Зато *Matlab* победил в быстром преобразовании Фурье.

Следует отметить, что измерение производительности зависит от целого ряда факторов и для достоверности требует многократных повторений. Наши измерения проведены лишь в целях относительного сравнения. Даже отличие в долях секунды может стать убийственным, когда объем вычислений становится больше или требуются тысячи повторений. Однако для небольших приложений или задач эти отличия производительности могут никогда и не возникнуть.

> **«*Matlab* предоставляет умопомрачительный набор математических функций.»**

Еще один плюс *Matlab* – множество пакетов расширений (Toolbox), доступных в виде дополнительных надстроек для базового продукта *Matlab*. Они покрывают специфические области применения, вроде обработки изображений, вычислительной биологии и экономического моделирования и анализа, и сберегают кучу времени, когда нужен немедленный результат без изобретения велосипеда. MathWorks, изготовитель *Matlab*, поддерживает в сети сообщество пользователей, где они обмениваются кодом и помогают друг другу. Хотя некоторые другие платформы предоставляют пакеты расширений в стиле *Matlab*, доступные варианты в *Matlab* отлажены лучше, чем имеющиеся для свободных платформ.



› **Желанная IDE *Matlab*: отметим редактор рабочего пространства слева и историю команд под ним.**

Дорогое удовольствие

Matlab – потрясающий продукт, и после его тестирования для обзора мы очень расстроились, обнаружив, что он поставляется по заоблачной цене и с ежегодным взносом после первых 12 месяцев. Использование *Matlab* вне работы или университета, вероятно, чересчур дорого для большинства людей. Многие малые компании сочтут стоимость *Matlab*, при наличии свободных альтернатив, неоправданной – конечно, если только они не разрабатывают дополнения или сервисы для *Matlab*. *Matlab* – очевидный выбор для платформы численных расчетов, если вы можете его себе позволить и вам не нужна открытая лицензия. Иначе уж лучше обойтись чем-то свободным.

LINUX FORMAT Вердикт

Matlab

Версия: R2008a

Сайт: www.mathworks.com

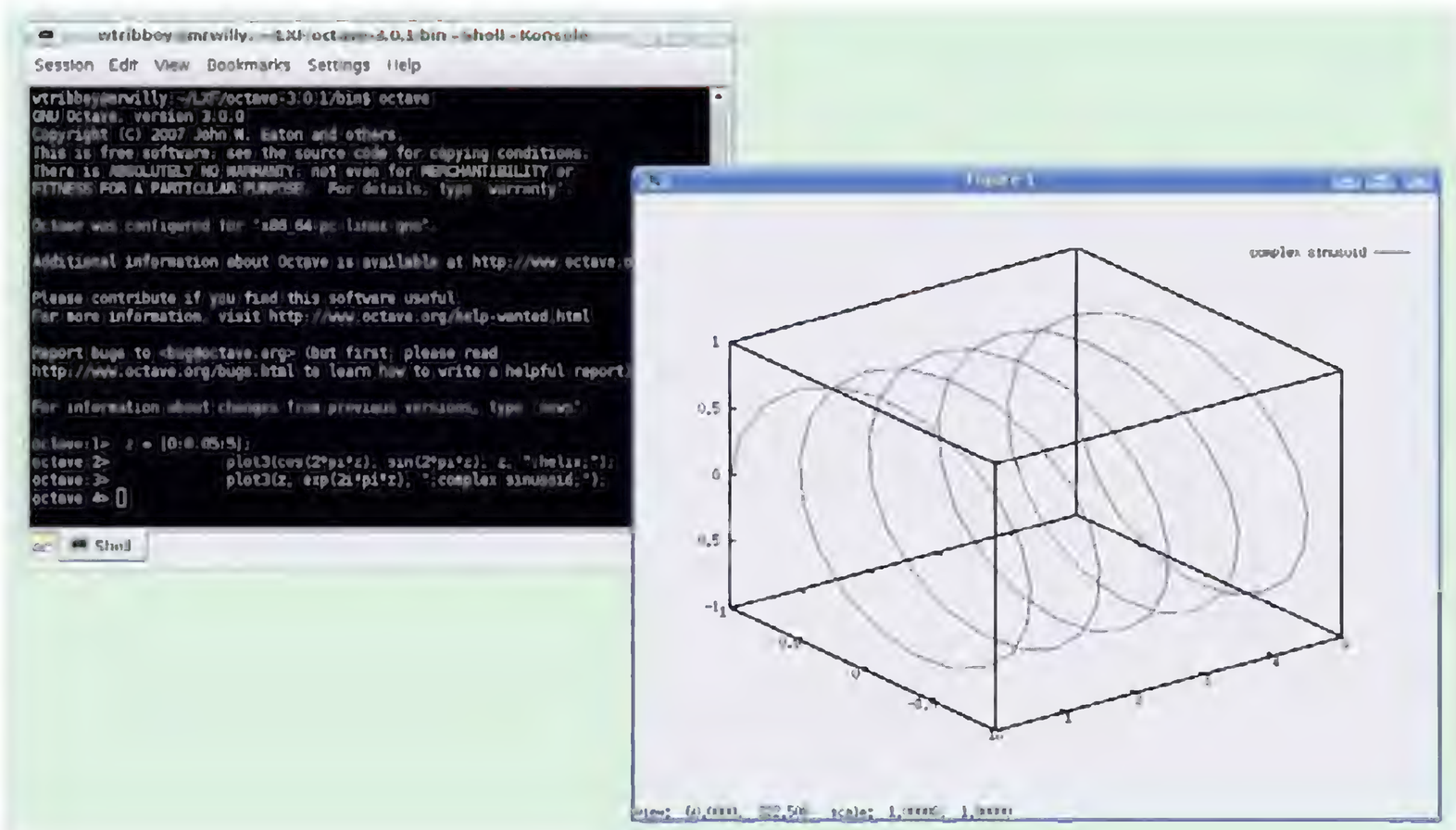
Цена: $1900 за первый год; $342 за последующие

» Matlab *великолепен, но может довести вас до нищенской сумы. Мы снизили ему баллы из-за дороговизны.*

Рейтинг 9/10

Octave

Естественно, есть GNU-альтернатива…



Начавшись как побочная идея работы инженеров-химиков в 1988 году, *Octave* вырос в сильную и недорогую копию *Matlab*, одну из наиболее известных свободных альтернатив *Matlab* с открытым кодом, и ныне старается добиться возможно лучшей совместимости с *Matlab*. Сайт *Octave* содержит подсказки по совместному использованию файлов пользователями *Octave* и *Matlab*.

На нашем тестовом ноутбуке *Octave* собрался и установился без сучка, без задоринки. Он запускается в окне оболочки, создающем ощущение некоторой пустоты. Однако, покопавшись, мы обнаружили на **sourceforge.net** проект *Octclipse*. Описание на сайте скудное, но, судя по названию и краткой справке, он основан на открытой IDE Eclipse. *QtOctave* – другая оболочка для *Octave*. Таким образом, вы можете заполучить IDE для вашего *Octave*, но это потребует дополнительных поисков и сборки. Однако, если одна IDE вам не понравилась, всегда можно попробовать другую.

Между базовым языком матриц *Octave* и языком *Matlab* имеются некоторые отличия. Для многих файлов программ совместимость – не проблема, но может стать таковой с добавлением объектно-ориентированного программирования в *Matlab*, если большинство пользователей возжелают эту функцию. В Wiki *Octave* имеется список функций последней версии *Matlab* (R2008a), отсутствующих в *Octave*; часть перечня, касающаяся объектов, содержит 50 пунктов. Если многие пользователи обзаведутся объектно-ориентированным *Matlab* – прощай, совместимость.

> «Проекты вроде *Octave* отзывчивее к запросам пользователей.»

Теперь о хорошем: сайт проекта приводит список уникальных функций *Octave*. Одна из них – определение функций прямо в строке [inline]. При работе в командном окне данная функция облегчает паковку повторяющихся шагов более выгодным способом, чем написание и сохранение файла программы. В *Matlab* этого сделать нельзя. Другая приятная функция – использование одиночных или двойных кавычек для определения строк, как делается в Python. Еще одно действие, выполнимое в *Octave*, но не в *Matlab* – использование символа # для комментариев. То есть, размещение **#! ./usr/bin/Octave -q** в начале файла программы приведет к тому, что оболочка выберет *Octave* для ее запуска; тут просматриваются интересные варианты использования при развертывании приложений. Эти небольшие отличия могут показаться незначительными, но, вероятно, они были добавлены по велению сообщества *Octave*. Проекты вроде *Octave* более отзывчивы к запросам пользователей, чем корпоративный элемент вроде The MathWorks.

Приличная скорость

Было бы весьма печально, если бы выбор между свободной и проприетарной платформами означал провал в скорости ваших вычислений. Но *Octave* – не тот случай: он показал 80% производительности *Matlab* в тесте SVD и около половины скорости в тесте FFT. При вычислении FFT разница измеряется сотыми долями секунды. Неясно, что внутри *Matlab* отличает его от *Octave* (а также *Scilab*) и позволяет выжать эти сотые. Обнаружение мелких отличий и их ликвидация не проще поиска иголки в стоге сена, так как они обуславливаются множеством факторов, вроде использования памяти, загрузки процессора в момент тестирования, структуры данных и т.д. Главное, что производительность *Octave* более чем адекватна для любой задачи разумного объема.

› В *Octave* приходится изощряться самим – здесь нет стандартной IDE.

На сайте **octave.sourceforge.net** вы найдете дополнительные пакеты для использования в *Octave*, являющиеся *Octave*-эквивалентами пакетов расширений (Toolbox) *Matlab*. Список весьма обширен, а охватываемые предметные области столь же многочисленны, что и для пакетов расширений *Matlab*. Широкий выбор специализированного кода, который можно получить и использовать без оплаты, в значительной степени обуславливает мощь свободного и открытого ПО. Для пользователей, ценящих свободу и выбор, *Octave* – образцовый проект.

Статистика

Если ваши интересы – в сфере статистических расчетов, возможно, вам стоит обратить внимание на проект GNU R, задуманный как свободное воплощение языка S, который разработан в бывшей Bell Laboratories (ныне Lucent Technologies). R для S – то же, что *Octave* для *Matlab*, но сосредоточен на статистике. R предоставляет удобные программы статистической обработки данных и построение статистических диаграмм и графиков, способных потребовать значительного кодирования даже на языке матриц вроде *Octave*. R можно найти по адресу **www.r-project.org**.

LINUX FORMAT Вердикт

Octave

Версия: 3.0.1
Сайт: www.octave.org
Лицензия: GPL

» Octave *предоставляет вам платформу с монолитным ядром, а дополнения вроде IDE требуют склонности к проектам «сделай сам».*

Рейтинг 7/10

Sage

Когда нужна серьезная математика.

Sage – это больше, нежели платформа численных расчетов. Да, она оперирует числами, но также и алгебраическими символами, и это весьма подходящая система для выполнения расчетов и аналитических вычислений, а также для исследований в области криптографии, теории чисел, графов и еще кое-чего.

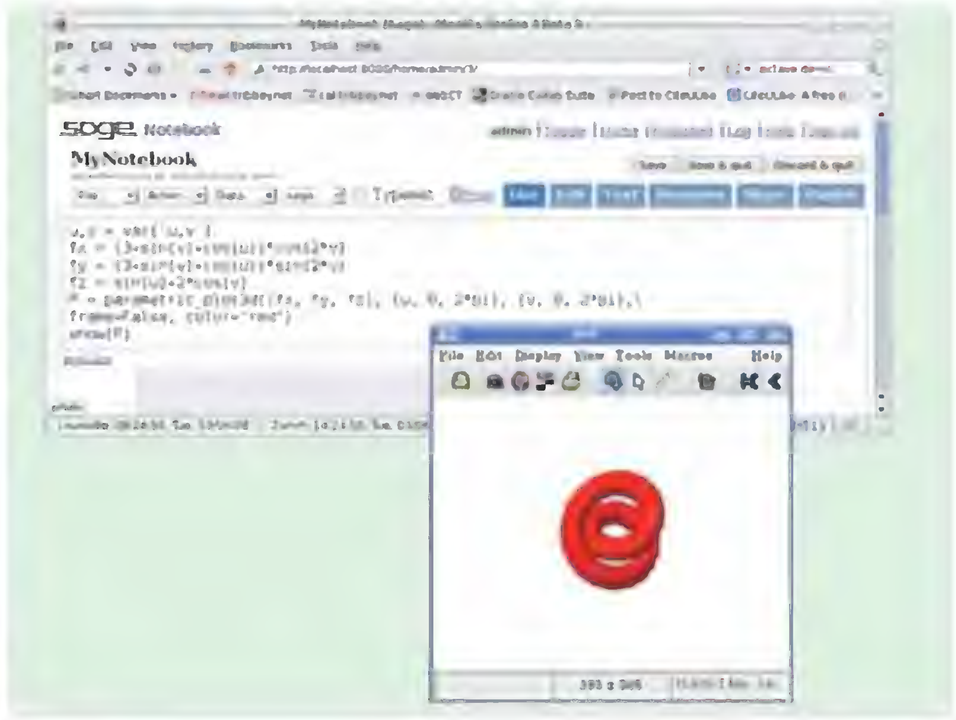
Sage несколько отличается от других платформ нашего обзора – тем, что использует Python, а не собственный язык программирования. Для некоторых пользователей это будет весьма привлекательным, поскольку если вы уже знаете и любите Python, то вам не нужно учить другой язык. В тестах производительности SVD и FFT, *Sage* (с использованием библиотеки Python Numpy) не уступает *Scilab* и *Octave*, причем в первом случае скорость была просто поразительной – с приличным отрывом от *Matlab*.

Sage – относительный новичок в данной группе, но использует приложения, родившиеся много лет назад. Например, *Maxima*. *Maxima* в виде своего предка системы компьютерной алгебры *Macsyma* изначально была разработана в Массачусетском Технологическом Институте в конце 1960-х. *Maxima* и другие компоненты *Sage* – это эклектичная смесь математического ПО.

Делиться надо

Одна из похвальных целей проекта *Sage* – организовать удобный обмен математическими идеями, а одним из способов достичь ее является использование текстового редактора с web-интерфейсом. Вы можете вводить команды в консоли, предоставляемой web-приложением *Sage*, а затем выполнять их. Это очень похоже на записи в математическом сетевом блокноте. Ваши блокноты могут быть сохранены или выложены для общественности, что дает множество позитивных последствий в обучении и совместных исследованиях. *Sage* можно попробовать без необходимости его сборки и установки, перейдя по адресу www.sagenb.org, зарегистрировавшись под свободной учетной записью и создав свой блокнот.

> «*Sage* – много больше, чем приложение численных расчетов.»



› **Графики в *Sage* обычно отображаются Java-апплетами. Неплохо для начинающего?**

LINUX FORMAT Вердикт

Sage

Версия: 3.0.1
Сайт: www.sagemath.org
Цена: Бесплатно GPL

» *Выбирайте* Sage, *если ваш математический двигатель требует высокооктанового топлива – просто приготовьтесь к изучению.*

Рейтинг 6/10

Euler Math Toolbox

Если вы хотите сохранить что-то для себя.

Euler Math Toolbox (EMT) заявляет о схожести с *Matlab*, но не желает быть его клоном. Он создан как математическая лаборатория, но после его использования у нас сложилось мнение, что это скорее персональный инструмент. Понятно, файлы блокнота *EMT* можно сохранять и публиковать, как любые другие, но мы бы не стали строить на них приложение.

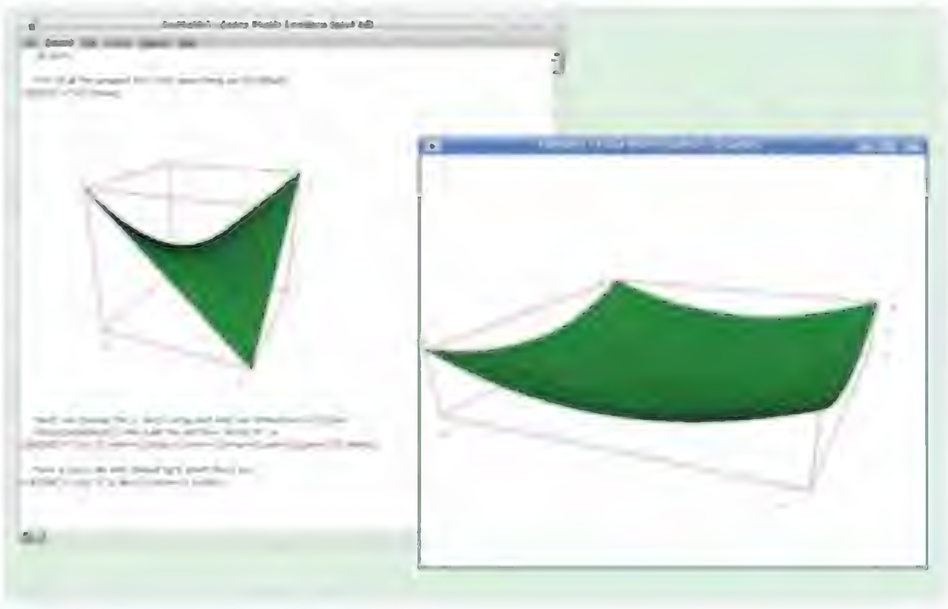
Кардинальное отличие *EMT* – то, что это не приложение Linux. Да, есть Linux-версия кода, из которой можно собрать весьма ограниченную редакцию *EMT*. Для желающих воспользоваться всеми возможностями *EMT*, на web-сайте рассказано, как запустить его под Wine. Интересное предложение – именно так мы и поступили. В версии *Wine* Kubuntu 8.04 *EMT* устанавливается без проблем. Вопрос в том, как его запустить.

Как вы могли предвидеть, производительность *EMT* при запуске в *Wine* не столь велика. Расчет SVD был вчетверо дольше, чем в *Scilab* и *Octave*. Тест FFT был выполнен не на матрице 500x500, а для вектора из 262 144 действительных чисел. Отметим, что это степень двойки; это говорит о том, что алгоритм БПФ не предназначен для размерностей более единицы, и что о создании массива данных с длиной, равной степени двойки, обязан позаботиться пользователь.

Светлая сторона

EMT заполняет определенную нишу, которую, вероятно, игнорируют *Matlab*, *Scilab* и *Octave*. Хотя *Sage* в какой-то мере может быть тем, чего ищут жаждущие простого инструмента для изучения принципов математики или для выполнения домашних работ, *EMT* особенно подходит для таких личных численных вычислений. Для символьных вычислений он использует те же функции Maxima, что и *Sage*. *EMT* – прекрасный выбор, когда вам необходимо что-то не столь всеобъемлющее, но все еще действенное.

> «*EMT* заполняет нишу, игнорируемую *Matlab*, *Scilab* и *Octave*.»



› **Простой и ясный интерфейс блокнота *Euler Math Toolbox* дает пользователям комфортный способ выражения идей.**

LINUX FORMAT Вердикт

Euler Math Toolbox

Версия: 5.5
Сайт: http://mathsrv.ku-eichstaett.de/MGF/homes/grothmann/euler/
Лицензия: GPL

» *Хоть и не столь исчерпывающ, как другие платформы,* Euler Math Toolbox – *прекрасный выбор для новичков.*

Рейтинг 5/10

Scilab

Scilab на одной прямой с *Matlab*.

Многие платформы численных расчетов в нашем обзоре были созданы и поддерживаются или отдельными лицами, или небольшой группой с мечтой о свободной альтернативе *Matlab* с открытым кодом. *Scilab* – пример платформы, уже несколько лет развиваемой усилиями большого управляющего комитета – Консорциума *Scilab*. Он включает в себя 25 корпораций и образовательных учреждений и управляется комитетом и научным советом. Вы можете думать об эффективности подобных структур что хотите, но данная команда великолепно поработала над *Scilab*. Территориально Консорциум *Scilab* находится в INRIA (Французском Национальном Институте Исследований в Компьютерных Науках и Управлении).

Среди всех рассмотренных не-*Matlab* приложений, *Scilab* подошел ближе всех к замене *Matlab*. Во-первых, хотя в *Scilab* нет того, что можно точно назвать IDE, окно команд связано с редактором, где можно разрабатывать и отлаживать функции. Многие действия, доступные в IDE *Matlab*, имеются или в окне команд, или в редакторе *Scilab*.

Он – не клон

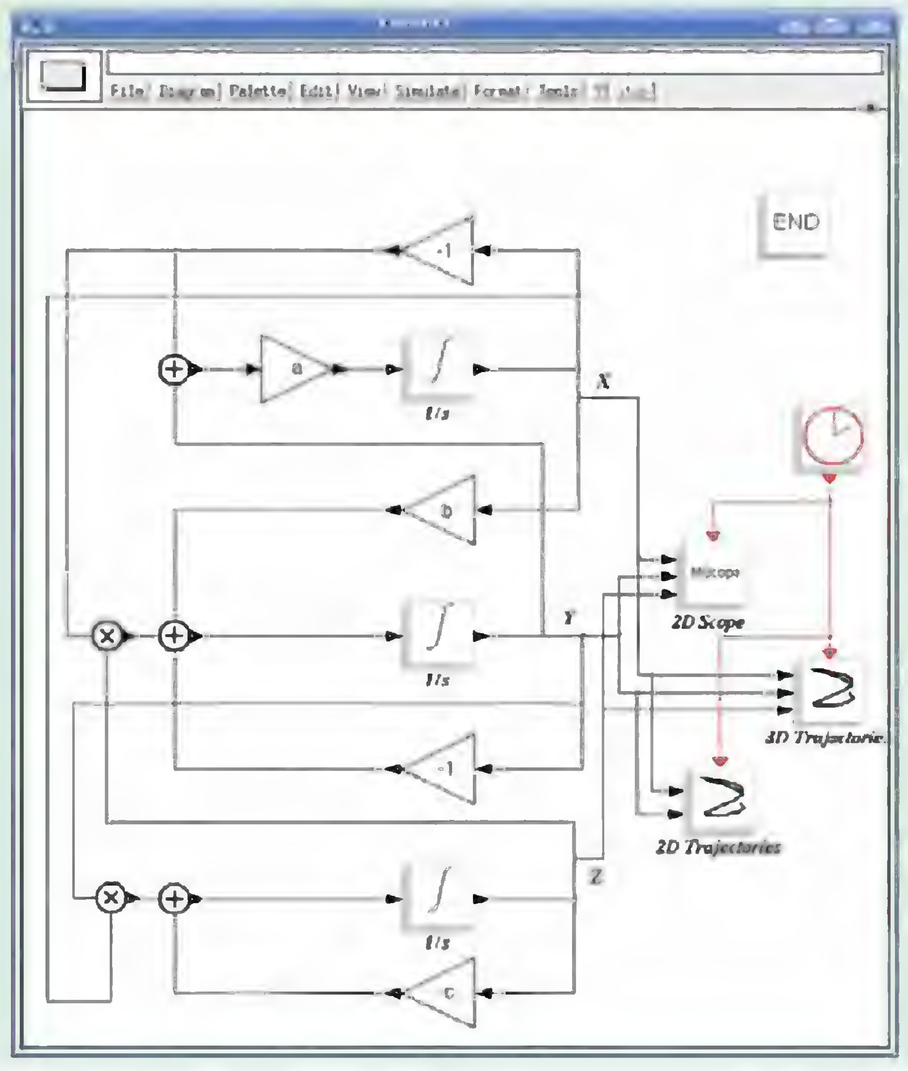
Следует также отметить, что *Scilab* не разрабатывался как клон *Matlab*. На страницах wiki *Scilab* есть раздел о процессе переноса файлов из *Matlab*, а в справке *Scilab* – раздел с советами по преобразованию *Matlab–Scilab*. Одна из страниц wiki *Scilab* заверяет, что портирование пакетов расширений *Matlab* в *Scilab* не потребует больших трудов.

Заглянув в справку на web-сайте *Scilab*, вы обнаружите не только впечатляющий список встроенных функций, но также несколько встроенных пакетов расширений (Toolbox). Они охватывают часть предметных областей, содержащихся в пакетах расширений *Matlab*, и описываются в справке отдельно от пользовательских, что дает повод считать их включенными в поддерживаемую кодовую базу, как и в *Matlab*.

Одно из расширений *Scilab* не имеет аналога в *Matlab*: это пакет Metanet для графов и сетей. Демо этого инструмента описывает планирование маршрута в парижском метро. Вы выбираете исходную и конечную станции, а инструмент отобразит линии метро, кратчайший маршрут вашего путешествия, а также список пересадок и общее время. Тем, кто изучает теории графов и сетей, этот инструмент может сберечь много времени.

У *Scilab* есть неожиданный козырь, отсутствующий у других соперников: Scicos, инструмент визуального моделирования. Scicos позволяет пользователю визуально создать систему, которую затем можно динамически имитировать. Это очень мощный инструмент, и его наличие в *Scilab* поднимает данный продукт над остальными платформами. Для среднего пользователя Scicos очень труден в изучении, но освоившись с ним, приобретаешь увеличение производительности и глубокое понимание разработки и моделирования систем. Scicos – свободный аналог продукта Simulink компании MathWorks, приобретаемого отдельно от Matalab. *Scilab* и Scicos, быть может, не столь блестящи, как *Matlab* и Simulink, но это серьезные конкуренты, которых стоит принять во внимание.

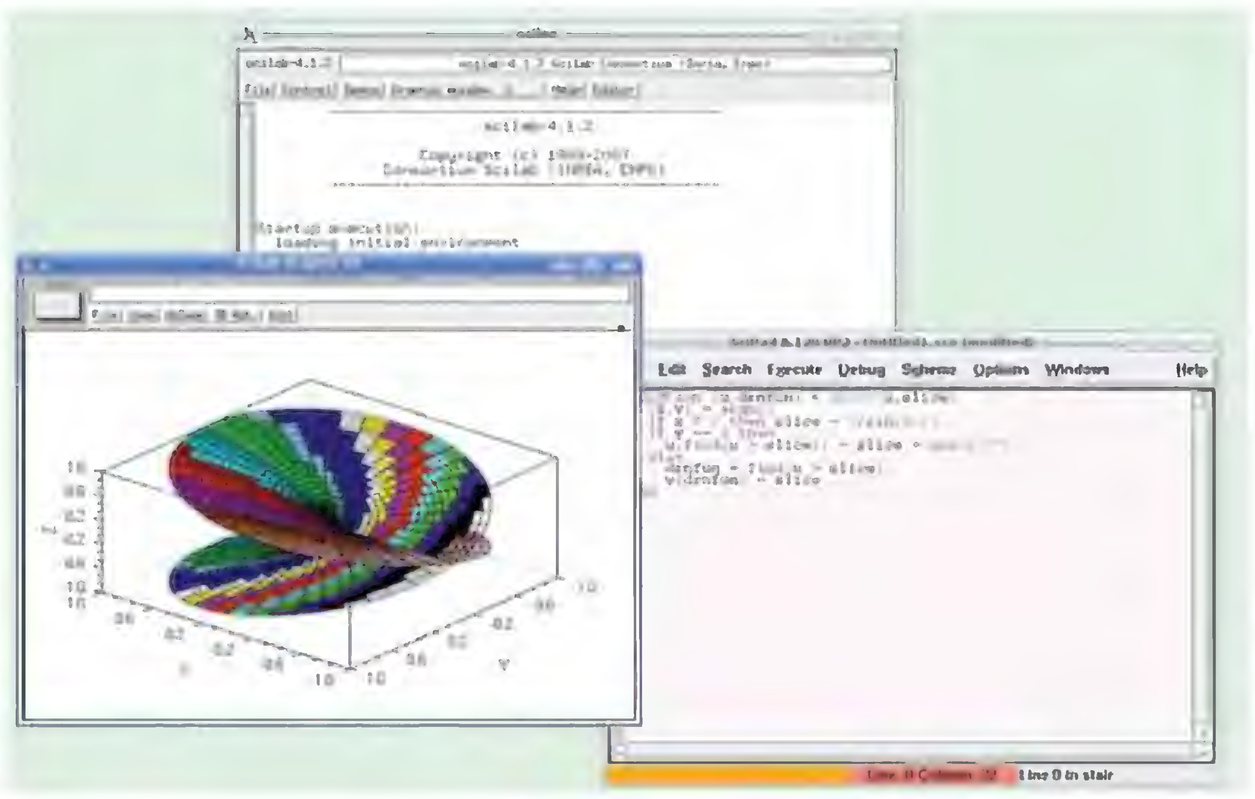
> «Козырь *Scilab* – Scicos, инструмент визуального моделирования систем.»



› Это модель аттрактора Лоренца в *Scicos*. Чтобы получить ее в *Matlab*, надо немало приплатить.

Великолепная пятерка

В этом обзоре использовалась версия *Scilab* 4.1.2. На момент написания, *Scilab* близок к выпуску долгожданной версии 5.0 – она объединит несколько важных архитектурных изменений, нацеленных на улучшение интерфейса пользователя. Мы загрузили ранний релиз-кандидат и обнаружили новый редактор переменных в меню окна команд. Он позволяет видеть определяемые переменные графически и легко изменять их значения. Похоже, эта возможность будет востребована. В архитектуру новой версии входит и Java: она используется в интерфейсе и взаимодействии с OpenGL.



› Редактор *Scilab* содержит многие функции IDE Matlab.

LINUX FORMAT Вердикт

Scilab

Версия: 4.1.2
Сайт: www.scilab.org
Цена: Бесплатно для некоммерческого использования
» Scilab – *хорошо увязанная платформа; ее возможности и функции значительно превосходят предложения собратьев.*

Рейтинг 9/10

Математическое ПО

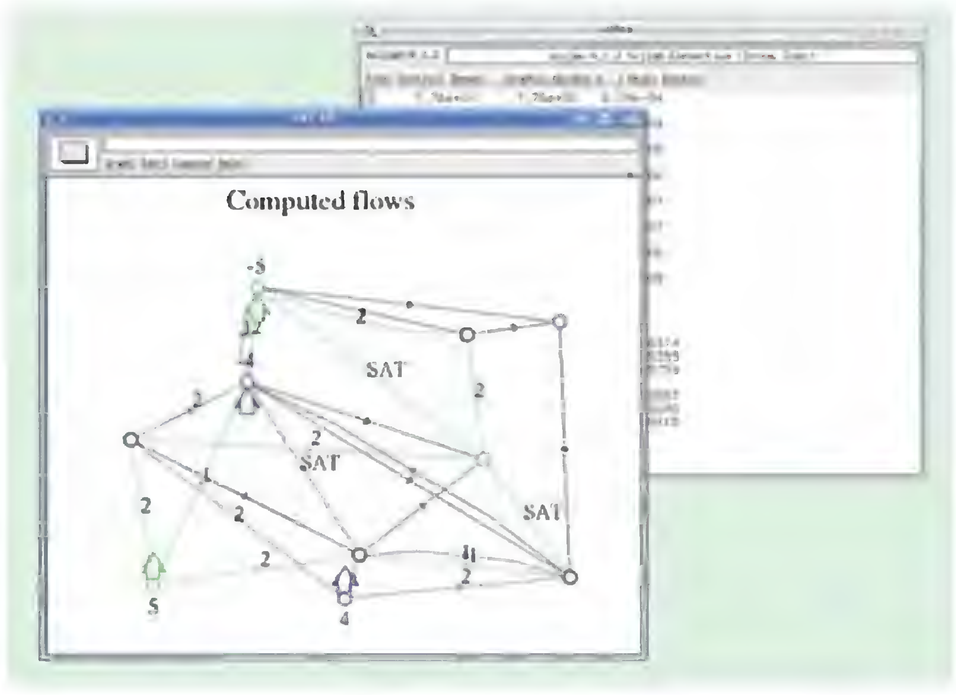
Вердикт

Scilab 9/10

Scilab содержит массу доводов в свою пользу: пакеты расширений, визуальное моделирование систем и имитацию, а также количество встроенных функций, которым легко найти соответствие в *Matlab* (мы имеем в виду число функций, а не их совместимость). *Scilab* подобрался к *Matlab* ближе, чем все остальные. Отчасти это может быть обусловлено тем преимуществом платформы *Scilab*, что она получает финансирование, которого нет у других проектов (а если есть, то гораздо скуднее). Благодаря глубине и широте своей функциональности, платформа *Scilab* достигает финишной ленты на несколько корпусов впереди ближайшего преследователя.

Отсутствие хорошей IDE в *Scilab* можно рассматривать как черный шар, но нас более заботит его не совсем свободная лицензия. Не похоже, что в версии 5 будет какая-то IDE, но в разделе для разработчиков сайта *Scilab* есть заявление, что версия 5 будет выпущена под известной свободной лицензией – может, все-таки GPL? Помимо улучшения архитектуры, смена лицензии будет большим шагом вперед.

Octave получает «серебро». Если вы закоренелый пользователь GNU или хотите быть частью проекта, в который легко сделать свой вклад, то не ошибетесь, выбрав прочную основу функциональности *Octave*, дополненную пакетами пользователей. Хотя для *Octave* имеется несколько IDE, проекту имеет смысл выбрать одну или создать свою, пусть даже и простую. Отсталая IDE всегда лучше, чем окно команд и текстовый редактор, но даже у этого мнения всегда будут противники.



Это демо пакета расширений *Scilab* для графов и сетей, используемый для решения задач потоков в сети. Мило, правда?

> **«*Scilab* версии 5 выйдет под хорошо известной открытой лицензией – возможно, GPL?»**

Обратная связь

Конечно же, кое у кого возникнет желание высказаться. Может, мы забыли одну из ваших любимых функций? Сделали некорректное замечание? Нет ли у вас приложения на одной из этих платформ? Для вас есть вещи поважнее, чем IDE? Пришлите нам ваше мнение о Сравнении по адресу letters@linuxformat.ru.

Приятный сюрприз

Sage и *Euler Math Toolbox* были включены лишь с точки зрения разнообразия и выбора, поскольку то, что одному – численная нирвана, другому – шестой круг ада. И тем не менее мы были приятно удивлены, обнаружив, что *Sage* – достойный конкурент не только в численных, но и в символьных вычислениях. Он выделяется как единственная платформа, использующая напрямую один из современных языков программирования, Python. Поскольку с помощью *Sage* вы можете создавать математические документы прямо в web, то будет уже неудивительно, если математические блоги станут темой следующего интернет-бума. И не забудьте: впервые вы прочли это здесь... LXF

Таблица свойств

Название	Лицензия	IDE	Язык	Доступность расширений (Toolbox)	Совместимость с Matlab[1]	Система визуального моделирования[2]	FFT[2]	Компьютерная алгебра	Блокноты
EMT	GPL	☒	Matrix	☒	☒	17.28+-0.490	0.411+-0.001	✓	✓
Matlab	Проприетарная[3]	✓	Matrix	✓	✓	3.438+-0.182	0.017+-0.003	✓	✓[5]
Octave	GPL	✓	Matrix	✓	☒	4.180+-0.182	0.018+-0.003	☒	☒
Sage	GPL	☒	Python	☒	☒	1.288+-0.338	0.093+-0.014	✓	✓
Scilab	Полусвободная[4]	✓	Matrix	✓	✓	4.242+-0.221	0.272+-0.004	☒	☒

[1] Поддержка какого-либо способа преобразования на/с языка Matlab.
[2] Значения после +- означают стандартное отклонение для 100 вычислений.
[3] Этот пакет идет с дополнительными лицензионными взносами.
[4] Бесплатно для некоммерческого использования – в других случаях внимательно читайте лицензию.
[5] Только для MS Windows. Обеспечивает интеграцию с *MS Office*.

Distrowatch

Ежемесячная сводка новостей дистрибутивов Linux.

СЬЮЗЕН ЛИНТОН

Сьюзен Линтон является владелицей и оператором tuxmachines.org.

Сделайте одолжение

Недавно Rad Hat покончила с судебной тяжбой, затеянной FireStar Software и DataTern, о нарушении патентного права, защищающего некоторые способы взаимодействия приложений через реляционные базы данных. Red Hat отрицала претензии, но участвовала в процессе расследования и урегулировала все без суда. О виновности Red Hat ничего не сообщается, но состоялись переговоры о защите компании, ее ведущих разработчиков и пользователей кода Red Hat от дальнейших преследований. Некоторые хвалили соглашение и похлопывали Red Hat по плечу за такую заботу о сообществе Open Source. А вот я не в восторге.

Кубок? Чайник?

Часть соглашения, касающаяся сообщества – то место, где Red Hat берет под крыло своих ведущих разработчиков и пользователей. О защите других дистрибутивов, их разработчиков или пользователей речи нет. Очевидно, что Red Hat и Fedora содержат множество ведущих проектов, но каковы шансы, что они нарушают законодательство? Что купила Red Hat и чем поступилась DataTern?

Да и не совершила ли Red Hat тот же смертный грех, в котором два года назад обвинялась Novell? Если Red Hat отвергла любые обвинения, но обошлась без суда, следует ли думать, что они виновны? Такое впечатление явно есть. Из-за этой кулуарной сделки о виновности задумается любой, как в случае с соглашением Novell, узаконившим претензии Microsoft. Радуйся, коли охота, дорогое сообщество, но я разочарована. Выигрыш в суде был бы победой всего Открытого Кода, а не только кода Red Hat.

susan.linton@futurenet.com

Ура Geeko!

OpenSUSE 11.0 Новый установщик, переписанный менеджер программ Zypp и передовые 3D-эффекты.

Выпуск OpenSUSE 11.0 стал «гвоздем сезона» в мире дистрибутивов. Данный релиз привнес множество изменений и внутри, и снаружи. Обновления заметны уже по новому продвинутому установщику и продолжаются далее, включая окно-приветствие, заставки и основные рабочие столы.

Да, это KDE 4 – а проблем никаких нет!

Инсталлятор обзавелся массой улучшений, экономящих время. Среди них опция «Использовать автоматическую настройку», применяющая стандартные параметры для ваших устройств; «Установка из образов» для внесения определенных пакетов в установочный образ для многих групп стандартных пакетов программ, например, рабочего стола Gnome; удален шаг определения пароля суперпользователя-root, в установщике теперь используется пароль первого пользователя.

Одним из основных окружений рабочего стола, включенных на образ DVD, является KDE 4.0.4. Версия для OpenSUSE поставляется с интуитивным и знакомым оформлением, повторяющим рабочий стол KDE 3. Многие приложения KDE портированы или переработаны для этого релиза, а миграция персональных данных показалась мне простой и безопасной.

> «Этот релиз портировал или переработал многие приложения для KDE 4.»

KDE 4 в OpenSUSE приятно стабилен и удобен, и на всем лежит отпечаток лоска, присущего команде SUSE. Это лучшее из виденных мной воплощений KDE 4, хотя я и отметила проблемы с *KNetworkManager*.

Также в релиз вошли последние стабильные версии рабочих столов *Xfce 4*, Gnome 2.22 и KDE 3.5.9, оформленные в том же зеленом стиле OpenSUSE, что и KDE 4.

Состав данного релиза: *Firefox 3.0b5* (обновляется до 3.0 через Интернет), *OpenOffice.org 2.4.0*, *GIMP 2.4.5, Inkscape, Pidgin, Liferea, Ekiga, GnuCash, Evolution, Tasque, KOffice, K3b, Brasero* и *Compiz Fusion 0.7.6*. Наслаждаться мультимедиа вы сможете с помощью *Banshee 1.0, Amarok 1.4.9.1, Totem* или *Kaffeine* – в дистрибутиве теперь можно легко, в два щелчка, установить необходимые кодеки и библиотеки. Внутри OpenSUSE 11.0 предоставляет Linux-2.6.25.5, *X.Org X Server 1.4.0.90, X.org-X11 7.3* и *GCC 4.3.1*.

Вы всегда можете добавить дополнительные приложения через менеджеры пакетов *Zypp* или *Yast*. *Zypp* времен серии 10.x полностью переписан, оснащен новым инструментом разрешения зависимостей, интегрирован с PackageKit и почищен от ошибок. Результат – намного большая скорость и более чистая работа для конечного пользователя. *Qt*- и *GTK*-интерфейсы доработаны как с внешней, так и с внутренней стороны. Мне показалось, что менеджеры управления пакетами – и графический, и в командной строке – демонстрируют значительное увеличение быстродействия. В моем случае *GTK*-интерфейс обновлений в Gnome рухнул, но я пока что не видела других подобных отчетов.

Несмотря на некоторые шероховатости, OpenSUSE 11.0 – крепко сбитый, интересный релиз. Если вам понравится версия с KDE 4, который в OpenSUSE 11.0 действительно работает, то этот дистрибутив для вас. И вообще, для релиза точка-0 он отлично выглядит!

www.opensuse.org

Свежее дыхание

Linux Mint Ярлык «прост в использовании» еще больше подходит новому MintTools.

Linux Mint стал одним из дистрибутивов, которые я рекомендую новым пользователям, из-за его совместимости с широким спектром устройств, полной поддержки мультимедиа и полезному набору системных утилит. Версия 5.0 была выпущена к общему восхищению и уже накопила багаж хвалебных обзоров.

Mint – восхитительно простой дистрибутив для новичков в Linux.

Linux Mint основан на Ubuntu Linux и еще улучшен. MintTools добавляет новый уровень управления; настраиваемое меню Slab намного проще в использовании, чем предлагаемые аналоги; и Mint совместим с репозиториями Ubuntu, открывающими целый мир приложений.

Коллекция *MintTools* содержит *MintAssistant* (где можно настроить пароль root), *MintBackup*, *MintUpdate* и *MintInstall* (устанавливает приложения из репозиториев Mint, Ubuntu, GetDeb). MintMenu в данном релизе претерпел незначительные изменения: добавились опции для настройки списка приложений.

Среди других полезных утилит – модуль Драйверов устройств Ubuntu (устанавливает проприетарные драйвера для ряда графических и Ethernet-чипов), *EnvyNG* (устанавливает графические драйвера), Драйвера беспроводных устройств Windows (импортирует драйвера с раздела Windows) и *Synaptic*, настроенный на репозитории Mint и Ubuntu.

Linux Mint поставляется также с *Firefox 3, Thunderbird, OpenOffice.org 2.4.0, GIMP 2.4.5, Pidgin, MPlayer, Totem, Rhythmbox* и *Brasero*. Это область, по-моему, действительно сияет, поскольку многие новые пользователи рассчитывают на полную поддержку мультимедиа прямо из коробки.

Некоторые говорят об увеличении быстродействия, но я этого не заметила. Внешне Mint остался прежним, так что это прекрасное обновление ОС, уже ставшей легендой. Если вы ищете простоты, возможно, Mint освежит ваш компьютер.

www.linuxmint.com

Вдали от Slack

Zenwalk 5.2 Улучшенное управление пакетами и поддержка мультимедиа.

Zenwalk основан на уважаемом дистрибутиве Slackware, но разработчики стараются сделать его доступным даже для зеленых новичков. Цели – быстрота, компактность и полнота. У меня уже несколько лет стажа приятной работы в Zenwalk, и разработчики, похоже, всегда находят пути улучшать его с каждым релизом. При возможности использования на старых компьютерах Pentium III всего с 128 МБ ОЗУ, он достаточно современен, чтобы покушаться на вашу супер-новую машину.

Хотя Zenwalk и основан на Slackware, ощущение от него совершенно другое.

Zenwalk упростил установщик Slackware, сделав его удобнее для новых пользователей, а при первой загрузке запускается несколько мастеров настройки (например, чтобы настроить пароль root и учетные записи). Этот релиз преподносит обновленный внешний вид, переписанный менеджер пакетов, улучшенную поддержку мультимедиа и множество исправлений ошибок.

Zenwalk поставляется с *Iceweasel 2.0.0.14, Icedove, MPlayer, Pidgin, AbiWord, Gnumeric, GIMP 2.4.6* и *Brasero*. Ядро – 2.6.25.4 с *X*-сервером 1.4.0.90 и *Xfce 4.4.2*. Вдобавок вы найдете такие программы, как *Amarok, Blender,* драйвера Nvidia, *libdvdcss* и KDE 3.5.9, доступные для установки через репозитории.

Инструмент управления пакетами *Netpkg* щеголяет графическим *GTK*-интерфейсом и древовидным списком пакетов с фильтрацией и поиском. Он менее интуитивный, чем, например, в *Synaptic*, но акклиматизация пользователей не займет много времени. Я устанавливала различные пакеты, включая группу KDE, и обнаружила, что все работает по-настоящему хорошо. Список зависимостей требует подтверждения, а затем все устанавливается без проблем.

Zenwalk вырос в прекрасный дистрибутив. Легче ли он в использовании, чем Slackware? Еще бы. Так же ли он прост, как, скажем, Linux Mint? Пока нет, но близок к этому.

www.zenwalk.org

Самые популярные дистрибутивы

Бывший редактор Linux-Watch Стивен Дж. Воган-Николс [Steven J Vaughan-Nichols] счел, что Xandros – самый популярный дистрибутив Linux, поскольку он развернут на многочисленных проданных Eee PC. Ну, а Джо «Zonker» Брокмейер [Joe Brockmeier], менеджер сообщества OpenSUSE, не согласился и опубликовал числа. Проблема в том, что большинство дистрибутивов понятия не имеют о численности своих пользователей. Вот несколько грубых приближений, которые я смогла получить от дистрибутивов* и из других источников.

Дистрибутив	Число пользователей
Ubuntu*	От 4 до 6 миллионов
Mandriva*	«несколько миллионов»
Fedora	2 миллиона
openSUSE*	1,5 миллиона
Xandros	1,1 миллиона
Debian	1 миллион
Slackware	500,000
Sabayon*	15,000

Хит-парад дистрибутивов

10 самых посещаемых страниц на DistroWatch.com со 2 по 30 июня 2008 (среднее число визитов в день)

Дистрибутив	Число визитов	
OpenSUSE	2,687	↑
Ubuntu	1,886	↓
Mint	1,477	↑
Debian	1,391	↑
Fedora	1,256	↓
PCLinuxOS	1,111	↓
Mandriva	743	↓
Damn Small Linux	700	↑
Dreamlinux	679	↑
CentOS	629	↑

» DistroWatch.com следит за популярностью дистрибутивов, основываясь на числе визитов на страницу каждого из них. Хотя это не дает представления о действительном числе установок, но показывает, какие дистрибутивы более популярны за определенный промежуток времени. LXF

ФСС идет

Установить открытое ПО, сэкономить на лицензиях, обеспечить информационную безопасность страны и побудить госструктуры к новым миграциям? Все это вполне возможно, полагает **Сергей Станиславович Ковалевский.**

В последние годы много говорят об использовании Linux и свободного ПО в государственных структурах. Германия, Франция и другие страны уверенно идут по этому пути, а что же Российская Федерация? Если судить по средством массовой информации, то в глобальном масштабе внедрение Linux в нашей стране ограничивается пресловутым «школьным проектом». На практике это не совсем так – просто многие из нас издавна привыкли делать свою работу молча. По заданию редакции Linux Format Виктор Федосеев встретился с Сергеем Станиславовичем Ковалевским – доктором технических наук, академиком РАЕН, Заместителем председателя Фонда социального страхования РФ – той самой организации, которая выплачивает нам «больничные», «детские» и прочие пособия...

LXF: Сергей Станиславович, говорят, что Фонд социального страхования (ФСС) переходит на свободное ПО? Хотелось бы узнать подробности...

СК: Позвольте мне сразу же кое-что прояснить. Говорить: «Вот, появился Linux, а почему вы решили на него перейти?» – это не совсем правильно. Еще в 1991 году я стал заниматься проблемой информационной независимости и безопасности, именно в это время началась активная компьютеризация служб Правительства РФ и Москвы. Можно сказать, что я и мои коллеги были одними из «пионеров» в решении данных проблем. Раньше о них вообще не принято было говорить, и только в последние годы на самом высоком уровне стали обсуждать вопросы информационной безопасности и открытого ПО. Ведь и тогда, и сейчас мы тратим огромные государственные деньги на развитие зарубежных технологий. Мы постоянно покупаем право (лицензию) на использование иностранного программного обеспечения, а как такового продукта у нас и

на рекорд

нет. Многие государства мира, оказавшиеся в подобной ситуации, всё больше интересуются программным обеспечением с открытым исходным кодом, ставшим в последнее время общепризнанным решением для правительственного сектора.

В России уже есть примеры отечественных разработок системного уровня, которые превосходят западные аналоги. Самым наглядным является десятилетний опыт успешной эксплуатации крупномасштабной информационной системы Фонда социального страхования РФ, а также информационные системы ряда структур Правительства РФ и Москвы. Это является доказательством реализуемости озвученной в ноябре 2007 года Президентом России Д.А. Медведевым идеи о необходимости перевода государственных структур на открытое ПО.

В настоящее время достигнута договоренность с руководством МИФИ о разработке учебного курса, в котором будут рассмотрены вопросы построения крупномасштабных распределенных информационных систем с использованием открытого ПО и российской СУБД *HyTech*. Разработка курса начнется в новом учебном году.

LXF: *HyTech*? Никогда о ней не слышал...

СК: О, это длинная история! В 1986 году мы взялись за компьютеризацию Онкологического центра и сели за персональные компьютеры. Сами понимаете: Советский Союз, никакого Microsoft с ее *SQL Server*. *Oracle* уже был и неплохо работал на мейнфреймах этак за полтора миллиона долларов, которые могли себе позволить западные клиники (да и то далеко не все). А что же мы? Поставить *Oracle* на ПК с 10 МБ оперативной памяти было, конечно, нереально. Перед нами встала задача: повысить производительность приложений, традиционно работающих на больших машинах класса ЕС, в 100 раз! И мы начали разрабатывать свои методы доступа, привязанные к PC, процессорам Intel, а не к мейнфреймам. И уже к 1989 году реализовали уникальные технологии, аналогов которым тогда не было. С помощью нашей программы на обычном ПК можно было обрабатывать сотни тысяч записей – по тем временам очень неплохо. Через два года по быстродействию к нам приблизились американцы с новой технологией «Rushmore» [Еще быстрее], которую потом купила Microsoft и включила в свой *SQL Server*.

Начало девяностых ознаменовалось компьютеризацией правительственных структур. И мы выполнили ее на нашей СУБД *HyTech*. Рынок СУБД в РФ тогда, можно сказать, только становился. Нам предлагали различные пакеты: *Clipper*, *Paradox*, *Fox Pro* и другие подобные системы. Я всегда спрашивал: «Зачем покупать западный продукт, исходных кодов которого нам никто не даст, а значит, мы не сможем его дорабатывать? Цена будет сумасшедшая – ведь потребуются специалисты, поддержка и много чего еще. Нам нужно свободное ПО либо отечественный продукт».

LXF: Понятно, а что было потом?

СК: Сегодня *HyTech* – это бесплатная отечественная система управления базами данных, существующая и для Windows, и для Linux (пока только в виде бета-версии). *HyTech* поддерживает распределенный по всей территории Российской Федерации, децентрализованный банк данных Фонда – кстати, именно поэтому вы не найдете ворованных баз ФСС на лотках рядом с другими. Чем на большее число кусочков разбита информация, тем ниже ценность каждого из них – а это, помимо прочего, возможность экономить, в хорошем смысле этого слова, на защите.

Что же касается базы данных, смотрите сами [*открывает в браузере Портал* – **fz122.fss.ru**]. У нас почти 190 000 зарегистрированных пользователей. Сегодня добавилось еще около тысячи, а к концу года будет все двести. Налоговики, служба судебных приставов... Всего более 100 ТБ данных. В ряде таблиц число записей превышает 300 млн., а самих таблиц в базе данных – несколько тысяч. И СУБД *HyTech* в реальном масштабе времени обрабатывает весь этот банк данных, используя OLAP-технологии. Кстати, одна из самых крупных баз данных (если не самая крупная) в Московском Правительстве управляется СУБД *HyTech* на 4-процессорном сервере Intel. При этом с базой данных одновременно (!) работает свыше 850 пользователей и выполняются 120–125 тысяч транзакций в день. Как видите, аппаратные требования весьма скромные.

LXF: Что ж, это действительно впечатляет. Думаю, мы еще вернемся к *HyTech* по ходу нашей беседы, а сейчас давайте поговорим о миграции. Насколько я понимаю, она находится в процессе осуществления? Какое открытое программное обеспечение использует Фонд сегодня?

СК: Мы активно применяем *OpenOffice.org*. Windows-версия установлена примерно на 20 тыс. рабочих мест, только на четверти из них имеется параллельная инсталляция лицензионного *Microsoft Office* – если это необходимо для обеспечения 100% совместимости по форматам документов с другими контрагентами из федеральных структур. Используем также *Samba*, web-сервер *Apache*, *Wine*, PHP, GCC и т.д. Из дистрибутивов – Debian, ALT Linux, FreeBSD. Изучаем на предмет возможного применения Mandriva. Предвосхищая ваш вопрос, скажу: не факт, что в конечном итоге Фонд выберет какой-то один конкретный дистрибутив. У каждого продукта есть свои сильные и слабые стороны: что хорошо подойдет для клиентского ПК, может плохо работать на серверах.

Вообще, у нас в организации приняты два правила. Первое: не использовать проприетарное ПО, если есть свободная альтернатива. И второе: использовать, по возможности, только отечественное ПО. Кстати, все программные продукты, приобретаемые региональными отделениями, должны быть согласованы с центральным аппаратом. Раньше мы давали добро на закупку платного ПО; сейчас, если регион запрашивает коммерческий продукт, у которого есть свободный аналог, заявка отклоняется. Поэтому, по сути, из проприетарного ПО на рабочих компьютерах сейчас установлена только операционная система (сами понимаете, явление времен-

ное), а также инструменты для обмена информацией с другими федеральными структурами (определенные соглашением архиваторы, средства защиты данных).

Департамент информационных технологий Фонда изучает существующее свободное ПО, проводит его тестирование на предмет соответствия нуждам ФСС и затем рекомендует его к использованию в исполнительных органах Фонда на рабочих местах специалистов в зависимости от выполняемых ими функций.

LXF: А какие, кстати сказать, у вас в организации компьютеры? Уживаются со свободным ПО?

СК: В Фонде нет дорогих серверов: мы используем архитектуру x86. Правда, есть быстродействующие RAID-массивы, многопроцессорные (до 4 процессоров) и кластерные (до 2 серверов в кластере) конфигурации. Такой подход был продиктован начальными условиями при построении системы и ставкой на быстродействующую отечественную СУБД *HyTech*, о которой мы уже говорили.

На рабочих местах специалистов используются ПК на базе процессоров Intel от P-III 700 МГц до Core 2 Duo 1,8 ГГц. Операционная система – пока что Windows (NT/2000/XP). Она поставляется предустановленной на новые компьютеры. Данное решение будет действовать на весь переходный период, т.е. до окончания миграции. Как только период закончится, в нашем госконтракте с поставщиком будет указан конкретный дистрибутив (или дистрибутивы), и мы начнем получать компьютеры с предустановленным Linux

LXF: Вот как? А что же тогда работает под Linux?

СК: В настоящее время на свободное ПО полностью переведена серверная часть Портала, предоставляющая в реальном режиме времени статистику обо всех аспектах деятельности Фонда на основе первичной информации из распределенного банка данных. Эта связка выглядит следующим образом: Linux/*Apache*/PHP/*HyTech*. Сервера, как я уже говорил, x86-е.

Многие ИТ-специалисты в настоящий момент работают сразу в двух ОС: Windows и Linux, установленных на одном компьютере. Пробуем строить контроллеры доменов на основе *Samba* и OpenLDAP (кстати, на этой задаче очень хорошо показала себя Mandriva) и т.д. – работы много.

LXF: Получается, переход сейчас в самом разгаре? А что еще вы собираетесь сделать?

СК: У нас запланированы следующие этапы:

» Установить *OpenOffice* для Windows на рабочих местах пользователей – с этим мы фактически уже покончили.

» Перевод серверов управления сетью (доступом, идентификацией, предоставления ресурсов и пр.) на Linux, управление сетью средствами Linux-серверов. Сюда входят:

» Сборка типовых стендов, отработка на них предложений и замечаний инициативной рабочей группы по Linux, подготовка документации и How-To. Подобное уже имеется в Чувашском, Орловском отделениях, в Алтайском крае;

» Обучение администраторов сети, в том числе и на основе стендов;

» Тиражирование и разворачивание решения в исполнительных органы Фонда.

» Проведение подготовительных работ по переводу баз данных на SQL-сервер *HyTech* (Linux-версия), перевод баз данных под Linux;

» Работа со специализированным программным обеспечением (ЕИИС «Соцстрах») в Linux под *Wine*.

» Работа со специализированным программным обеспечением (ЕИИС «Соцстрах»), разработанным для платформы Linux.

» Обучение соответствующих ИТ-специалистов на всех этапах.

LXF: А когда вы планируете завершить переход?

СК: Я думаю, в конце этого – начале следующего, 2009 года.

LXF: А по каким причинам ФСС сделал ставку на открытое ПО? Я понимаю, что для вас лично это вопрос давно решенный, но Фонд – весьма крупная и распределенная структура.

СК: Пожалуй, первая причина – экономия бюджетных денег на лицензиях. Первоначально, *Microsoft Office* закупался не на каждое рабочее место, а только там, где это было необходимо: большинство специалистов не работали с неструктурированными документами, а использовали встроенные отчетные формы рабочих подсистем. С ростом компьютерной грамотности сотрудников возникла необходимость в оснащении офисным пакетом практически каждого рабочего места. Анализ показал, что для этого потребуется более 100 млн. рублей.

Вторая причина – это высокая стоимость централизованного управления сетью в продуктах Microsoft. К сожалению, лицензионная политика данной компании в части доступа к серверам (CAL-лицензии) делает централизованное управление довольно дорогим удовольствием. Например, чтобы развернуть у нас Active Directory, необходимо понести разовые расходы в размере около 500 млн. рублей, и при пятилетнем сроке замены оборудования (а за это время меняется и серверное ПО, которое уже не устраивают CAL-лицензии предыдущей версии) – это еще примерно по 100 млн. рублей в год. Также привлекает возможность контроля за безопасностью открытых программных кодов – про это я уже говорил выше; и, наконец, возможность отдохнуть от всевозможных проверок.

LXF: Вы уже успели столкнуться с какими-либо трудностями на этом пути?

СК: Самая главная трудность, которую мы видим сейчас – это создание родных Linux-приложений, автоматизирующих деятельность Фонда; сейчас они существуют только под Windows. Всего их свыше 40, функционал очень широк и меняется достаточно часто в соответствии с изменениями в законодательстве, поэтому переписывание их «с нуля» под Linux может стать большой проблемой. В настоящее время используем приложения в тестовом режиме, запуская их в *Wine*.

Еще одна проблема, имеющая касательство к быстрому изменению законодательства – унаследованные данные. Чтобы сравнивать результаты текущего года с прошлыми периодами и проводить анализ, документы надо фактически перенабирать вручную, а это очень трудоемкий процесс.

Из решенных трудностей можно вспомнить задачу оперативного анализа данных по всей России (финансовый контроль), когда возникла необходимость работы с распределенным банком данных. Первоначально она была решена на базе *Microsoft Internet Information Server*, но при росте количества одновременно опрашиваемых БД возникли проблемы со стабильной работой многопоточного CGI-приложения.

В результате, вторым шагом стала связка Windows/*Apache*/PHP/*HyTech*. Основным аргументом была стабильность системы, вызванная во многом более эффективным использованием ресурсов, которая, в свою очередь, в качестве приятного бонуса дает еще и повышение производительности. Windows выбрали, поскольку свободные серверы под управлением *Windows Server* уже были.

Связка Windows/*Apache*/PHP/*HyTech* удовлетворительно работала около года, пока в процессе расширения возможностей портала и подключения сторонних пользователей (страхователей, медицинских учреждений, налоговой инспекции и пр.) мы не наткнулись на странное поведение набора Windows/*Apache*/PHP. Периодически фрагментация памяти приводила систему в состояние клинча, требовалась перезагрузка. К этому времени портал уже имел 160 тыс. пользователей и представлял из себя кластер из балансировщика и трех фоновых серверов, обслуживающих порядка 20 тыс. запросов к СУБД *HyTech* в час. На мозговом штурме решили, что дешевле заменить Windows на Linux, чем выяснять, как бороться с фрагментацией памяти в Windows. В тот момент мы горели желанием

посмотреть на наших оппонентов – поклонников Windows: как бы они разобрались в «исходных кодах Windows, которые переданы в Россию», взяли бы, да и исправили эту досадную ошибку, приводящую к фрагментации памяти.

В итоге мы собрали Debian и поставили его вместо одного из серверов. В балансировщике ему дали ту же загрузку, что и на остальные. После этого сравнили загрузку системы – в Debian это было 7% против 60% в Windows! То есть производительность выросла почти на порядок! Таким образом, судьба Windows на серверах портала была предрешена. Сейчас весь кластер работает под Debian и легко обслуживает 250 тыс. пользователей, отрабатывает свыше 50 тыс. SQL-запросов к СУБД *HyTech* в час, при этом серверы загружены максимум на 30%.

LXF: Ну, мы-то всегда знали преимущества Linux как серверной ОС! А на чем написаны Ваши специализированные программы?
СК: На Borland Delphi.

LXF: Lazarus не пробовали?
СК: Пробовали, конечно – как раз Lazarus сейчас и изучаем. Недостаток – почти никакой документации: берем Delphi и сравниваем по аналогии. Подумывали также использовать *Qt*, но она не подходит по лицензии.

LXF: Какое влияние, по-вашему, окажет переход на простых пользователей?
СК: Я думаю, потребуется первоначальная адаптация, в основном к другой последовательности пунктов в меню ОС. В целом, пользователи не должны заметить перехода с одной платформы на другую – используемые ими программы останутся прежними или с прежним интерфейсом.

LXF: А что говорят администраторы?
СК: Для администраторов в процессе перехода работы, конечно, прибавится: в основном, это выслушивание жалоб и комментариев пользователей, оказание им помощи в адаптации в новых условиях, в сопоставлении функций прежнего привычного окружения с аналогичными, а может быть – и не совсем аналогичными функциями нового окружения. Также нередко руководством на администраторов возлагаются «полицейские» функции по отслеживанию фактов несанкционированной установки пользователями несвободного ПО и ведению паспортов на состав ПО, установленного на компьютерах.

Поначалу администраторам придется сопровождать параллельно несколько различных ОС и прикладных программ: в производстве останется часть компьютеров с Windows и *Microsoft Office* для обеспечения все той же 100% совместимости с форматами других федеральных структур, как вышестоящих, так и производящих информационный обмен с Фондом. В связи с этим видится увеличение трудозатрат для ИТ-подразделений.

С другой стороны, уже упомянутая мною странная политика Microsoft в части CAL-лицензий (на подключение к серверу пользователей или устройств) делает централизованное управление сетевыми ресурсами довольно дорогим удовольствием. Поэтому переход на свободное ПО в перспективе уменьшит трудоемкость обслуживания сети.

LXF: Ну а вы, как представитель руководящего звена, чего ожидаете?
СК: Независимости от поставщика (владельца прав) операционной системы и снижение стоимости информационных систем (экономия на различных лицензиях). Повышения расходов на оплату трудозатрат и поддержку (по крайней мере, на начальном этапе) в связи с внедрением и сопровождением. В остальном не изменится ничего.

LXF: Что ж, понятно. Но это, в каком-то смысле, планы на ближайшее и отдаленное будущее – а кто и что выиграл от перехода уже сейчас? Или пока это «работа, работа, работа»?
СК: Нет, почему же. Выиграли администраторы, и очень много: моральное спокойствие в связи с исключением ситуации использования нелицензионного ПО и с отсутствием личной ответственности за его применение отдельными сотрудниками (при условии паспортизации ПО на рабочем месте). Выиграл Фонд – в части более оперативного представления статистики о своей деятельности и, следовательно, большей прозрачности в работе. Выиграло движение Open Source, т.к. Фонд является далеко не последним по роли из федеральных структур, а что касается новизны применяемых технических и программных решений, то, наверное, одним из ведущих в Российской Федерации. К тому же в Фонде создана централизованная иерархия управления, поэтому масштабное разворачивание на его «территории» открытого ПО, несомненно, должно дать мощный импульс внедрению открытого ПО в федеральных структурах.

Фактически, когда тема Linux только начинала набирать популярность, скептики говорили: «Ну да, на Linux можно перевести какую-нибудь небольшую структуру: офис, бухгалтерию и т.д. Но сможет ли Linux потянуть по-настоящему крупномасштабную разнородную систему?». Сеть Фонда – одна из самых крупных в стране: она насчитывает свыше 26 000 компьютеров. И когда у нас все получится, у наших оппонентов не останется никаких аргументов, мы создадим прецедент и покажем: да, такое возможно на федеральном уровне. Ну, а там настанет черед других государственных структур.

LXF: С технической стороной вопроса понятно. А как вы решаете проблемы социального характера? Вы сами отметили: незнакомый интерфейс, непривычные программы и прочее?
СК: Для простого пользователя главная проблема – это привычка. Практически все программы имеют красивый оконный интерфейс. По мере освоения приемов работы с тем или иным ПО (некоторого времени адаптации) проблемы возникают все реже. Для примера можно сказать, что с 1998 по 2002 год Фондом регулярно проводились курсы обучения специалистов работе с продуктами Microsoft: *Word*, *Excel*, *PowerPoint*, *Outlook*, *Internet Explorer*, управлению файлами и папками в проводнике Windows... После перехода на *OpenOffice.org* пользователи «поворчали» и продолжили трудиться практически без переобучения. LXF

» Рубрику ведет
Евгений Балдин

» **Кто**
Владимир Сергеевич Житомирский, начальник отдела особых операций и председатель совета директоров Правового бизнес-бюро «Граф Маевский» (www.gm-legal.com).

» **Откуда**
Санкт-Петербург.

» **Проект**
Повсеместное использование свободного программного обеспечения.

» **Результат**
Контрагенты стали уважать ещё больше.

Непростые юристы есть в Санкт-Петербурге

Люди всегда неосознанно тянутся к комфорту; обычно подобное входит в конфликт с эффективностью. Возможно ли объединить оба этих устремления? **Владимир Сергеевич Житомирский** на своём примере показывает, что да.

По складу характера я очень любознательный. Люблю разбираться в сути процессов, которые мне интересны. Когда лет двенадцать тому назад стал пользоваться компьютером, то сразу понял, что это отнюдь не только пишущая машинка с экраном, но и что-то гораздо большее.

У меня есть приятель, выпускник Можайки, сделавший компьютеры своей профессией. Я высказывал ему свои пожелания, а он мне всё настраивал. Очень удобно, но довольно часто получалось так, что он делал не совсем то, что мне хотелось. Когда же он по моей просьбе объяснял свои действия, то я понимал лишь четверть сказанного, что мне очень не нравилось. В итоге я купил кучу книг, залез в Интернет и приступил к разбирательствам. Влез (опять же из любознательности) достаточно глубоко, нашёл и поставил кучу интересных программ. Всё работало наконец-то так, как мне этого хотелось.

Последствий от этого шага было три. Первое, самое приятное и важное: я, пусть с краю, но познал мир разнообразных программ; пусть через глоток, но ощутил вкус этого безбрежного океана. Второе – я вдруг понял, что приходящий на тот момент в офис «сисадмин» понимает в своём деле меньше меня. И третье – я перестал просить приятеля о помощи. И он, придя как-то в гости, полез было к нашим машинам, но был посрамлён. Тогда он произнёс историческую фразу: «Ну, с Виндой ты как-то ковыряешься по-дилетантски. А вот мастера, в отличие от дилетантов, работают в Линуксе». На мой вопрос, что же такое «Линукс», он загадочно пожал плечами. Это случилось где-то пять лет назад.

После этого разговора я, само собой, полез искать Linux. И понял, что он мне сейчас и без подготовки совсем не по зубам. Зато я открыл для себя целый мир свободных программ, в том числе *OpenOffice.org 1.x*, браузеры, почтовики и т.п. Долго пользовался *Firefox*, но в конце концов пересел на *Opera*. Не счесть, видимо, людей, которые из-за меня теперь пользуются этими браузерами. Я уж не говорю о нашем офисе и домашних компьютерах.

И вот в конце 2006 года мне попалась книга В.Г. Соломенчука «Linux. Экспресс-курс» (издательство BHV-СПб, ISBN 5-94157-548-3). Я прочитал её несколько раз и в конце концов начал что-то понимать. Пошли другие книги, наши и переводные. Перепробовал разные дистрибутивы: Mandriva, SUSE, ASPLinux, ALT Linux. Остановился на последнем, причём не из-за того, что он «лучший», а потому, что я в нём достаточно разобрался. Опять же, пользователи только русскоязычные, так что проще что-то искать.

До всего пришлось доходить своими силами. Поскольку дистрибутив был куплен, то я пару месяцев гонял его со всех сторон, копил вопросы. Потом зарегистрировал, получил 2 месяца вопросов и старательно их задавал. В основном отвечали нормально, но как-то раз сказали что-то вроде «читайте документацию». Я, правда, так ответившего тут же «обидел», что ответ сразу же получил. С поддержкой всё-таки беда. Это как раз то, что тормозит распространение свободных программ. Тот же ALT Linux предлагает поддержу на таких условиях: абонемент – 120 тыс. в год, разовые обращения – 650 руб. в час, минимально – 4 часа. Разве это под-

держка для обычного пользователя? Я даже не о деньгах: представьте, что кто-то приходит и что-то делает в вашей машине, ничего при этом не объясняя и, естественно, оставляя мусор после своих исканий. Кстати, когда я разбирался со звуком, то получил столько указаний по насилию над системой, что потом пришлось её переустанавливать; сразу вспомнился Билл Гейтс. И только после многодневной переписки меня попросили прислать вывод запущенного в терминале файла, после чего предложили поставить два пакета, и всё заработало. Мне кажется вполне естественным, что производители должны поддерживать свои продукты. Более того, они должны окружать заботой как раз новичков, потому что именно новички – это их будущее. А они, при всех их декларациях, не понимают до конца, что обычному миру и обычным людям тоже нужны свободные программы. Они ведут себя как шофёры начала 20-го века – кастовость и исключительность. Как же так: «допустить обычных людей к рулю и перестать от них отличаться?» Сразу оговорюсь, так думают далеко не все и не всегда, но проблема существует. Маленькие фирмы были бы панацеей, но где взять специалистов? Так что всё сами, с книгами Соломенчука, Орлова, Ди-Анн Лебланк и т.д. и т.п.

В результате компания работает на Linux. Претензий нет, хорошие и комфортные Интернет и почта, устойчивость системы, прекрасный офисный пакет. А для продвинутых ещё и неограниченный набор замечательных программ-помощников.

Мы не сильно крупные – у нас всего 7 машин, и две из них пока ещё двухоперационные (из-за принтера и сканера). Сейчас, вроде бы, проблема решена. Вот спадёт обычный для середины лета ажиотаж, уберём Windows совсем.

Используется, само собой, офисный пакет (*OOo*), *Firefox* и моя любимая *Opera*, почтовый клиент *Claws Mail* (кстати, хорошая замена *The Bat!*), штатные программы мультимедиа и просмотра. Любители (я в том числе) используют *GIMP* и *Inkscape*. Через *Wine* от компании Ethersoft стоит правовая система ДЕЛЬТА (www.referent.ru) – на наш взгляд, она удобнее Кодекса и Гаранта; там же установлены маленькие отечественные редакторы изображений: пятна, красные глаза и т.п. Довольно долго *Inkscape* не казался достойной альтернативой *Corel*, но потом, во-первых, научились работать, во-вторых, программа толково обновляется. Так что замену всему возможному с избытком нашли. Да, следует упомянуть, что в игры не играем. Правда, мои дети что-то ставят через тот же *Wine*.

До повсеместного перехода на Linux где-то с полгода стоял *OOo* (наряду с *Microsoft Office*), и было введено правило, что все внутрифирменные документы идут только в свободном пакете. Так и привыкли потихоньку.

У меня дома сейчас четыре компьютера: два на Linux, один двухоперационный и один на стареньком Windows 2000. В итоге моих увлечений наши дети (двенадцать, девять и шесть лет) пользуются двумя ОС, понимая разницу между ними. И ноют, требуя поставить Linux на последнюю машину. Они не компьютерные фанаты – они просто на них рисуют, пишут, слушают музыку и смотрят фильмы, немного играют и стараются всё понимать, как пользователи, знающие, что они хотят, но не более того. Очень смешно, когда младшая говорит: «Мама, сколько лаз тебе надо говолить, сто диск сначала надо отмонтиловать!».

Одно – это поставить цель и достигнуть её, а совсем другое – понять: что именно мы получили? Какие преимущества теперь у нас есть после перехода на Linux? Непростой вопрос. С деньгами всё ясно, а с остальным? Боюсь быть банальным, но чувства устойчивости, безотказности и «многопрограммности» системы важны для обычного пользователя, и именно это мы сейчас и ощущаем. Что касается системного администратора, то приходит в голову пример, как если бы механика из мастерской по ремонту «Москвичей» перевели в салон по ремонту, скажем, «Ауди». Сложнее, ответственнее, но удовольствие! Но престиж! Но деньги! Владелец или высший руководитель видит то же, что и пользователь, плюс весомая экономия не только средств, но и времени, и сил, плюс надёжность лучей бизнеса, касающихся ИТ в самом широком смысле этого слова, плюс удовольствие руководить умными людьми (мне кажется, что работа со свободным ПО развивает если не ум, то мышление).

Самое, казалось бы, нелогичное – это то, что у контрагентов использование нами Linux вызывает боязливое уважение и усиливает желание сотрудничать. Мы давно пережили время, когда хватались за каждого клиента. Теперь все наши контрагенты – весьма приличные и умные люди. Они умеют слушать и понимать, они видят изящество решений, и именно непохожесть на других их как раз и привлекает.

› Наталья Анатольевна Житомирская, генеральный директор. Ей принадлежат крылатые слова, которые мы используем в пропаганде (заменив марки): «Я хорошо помню свои ощущения, когда в 2001 году пересела с «девятки» на «Ауди». Я вновь их испытала, когда в январе 2008 года перешла на Linux».

Естественно, временами они задают каверзные вопросы о совместимости файлов и так далее. После выхода Vista мы провели много экспериментов. Выяснилось, например, что *OOo* (у нас версия 2.3.1 в обработке ALT Linux) открывает файлы .docx, в половине случаев и с внедрёнными диаграммами, а все «довистовские фокусы» – тем более. Как минимум, из попавшего к нам в руки. Хочу отметить, что .docx не открывается и в иных версиях родного офиса.

Обычно, говоря о свободных программах, говорят о свободе. Мне кажется, что рядовой пользователь скорее осознаёт не свободу, а полученные возможности. А их у свободного программного обеспечения немало, и скорее всего именно поэтому кое-кто из наших контрагентов тоже начинает использовать свободное ПО, правда, на ОС от Билла Гейтса. Но это только начало. Хочется отметить ещё один феномен. Люди в мире свободных программ удивительно доброжелательны и терпеливы, хотя есть и исключения, весьма редкие, к счастью. К людям я не отношу клоунов на многих форумах. Ну и, естественно, посильную помощь в агитации в продвижении свободного ПО оказывают работники ОБЭП…

Довольно длительное время мы обходились без системного администратора – со всем управлялись своими силами. Но так получилось, что наша компания патронирует СО «Гвардия» (армейский рукопашный бой). И один из его молодых сотрудников, Вячеслав Морусов, заканчивающий институт, увлёкся Linux'ом, естественно, не без моего участия. Сейчас он даже соответствующий диплом пишет: «Разработка программного комплекса с открытым кодом для min-POS систем на основе GNU/Linux» И как-то стихийно он объявил себя нашим сисадмином. Теперь мы вынашиваем планы по продвижению свободных программ к обычным людям и корпорациям. Конечно, при этом мы собираемся использовать, в первую очередь, свой инструментарий. И ищем на этом пути соратников и единомышленников.

Взгляд с другой стороны

Сколько людей – столько и мнений, и выше мы познакомились с точкой зрения пользователя. А теперь **Вячеслав Юрьевич Морусов** поделится видением происходящего со стороны, помогающей этому пользователю.

Мне не нравится называть себя сисадмином, так как мой круг интересов к ПО выходит за рамки требований, предъявляемых к этой профессии. Мне не хочется поддерживать уже готовую систему, в которой ничего нельзя изменить – меня интересует развитие и улучшение отработанных узлов своими силами и силами других людей, которые более компетентны в ряде вопросов. Конечно, только моих знаний недостаточно, но наших с Владимиром Сергеевичем усилий хватает, чтобы заинтересовать в этом и других специалистов.

Идея о совместной работе с Владимиром Сергеевичем зародилась достаточно давно, и не на фоне каких-то проблем с компьютером. Я вёз команду на соревнования по рукопашному бою в Архангельск, а одним из участников сборной был его сын. Владимир Сергеевич был сопровождающим. Мы сидели в вагоне-ресторане и делились мнениями о свободных программных продуктах. Мне на тот момент не хватало опыта в бизнесе, а ему – технических знаний. Вот так появилась идея объединить наши усилия в продвижении открытого ПО.

Причины моего тяготения к свободному программному обеспечению примерно такие же, как у Владимира Сергеевича: предсказуемость, контроль и возможность получить желаемое. Когда вы покупаете машину, вас она полностью устраивает? Разве вам не хочется сделать её быстрее и уютнее? Хотя бы коврик постелить или чёртика на стекло налепить. А представьте теперь, что производитель лишает вас этой возможности, да ещё и вынуждает вас покупать раз в несколько лет новую, потому что отказывается обслуживать старую, и не даёт никакой возможности выбирать сервис. Возникнет ли у вас повторное желание обратится в эту компанию? То же происходит и с продуктами Microsoft. Вот только недостаточная информационная осведомлённость и ошибочное мнение, навязанное нам «программным гигантом», не даёт сделать шаг в сторону других вариантов. Причём это всё прикрывается заботой о «рядовых пользователях» и рассуждениями о том, что они самостоятельно выбор сделать не могут и поэтому их надо его лишить. Живой пример, что это не так – Владимир Сергеевич, который и есть тот самый рядовой пользователь. Объём его знаний в компьютерах был не больше, чем у любого другого человека, использующего ПК в повседневной жизни, просто он чуточку любознательнее. Сейчас для него не стоит вопрос, как установить драйвер к принтеру или сделать еще что-то, требующее вмешательства в ОС. Вот вам и подтверждение, что не нужно быть «компьютерным гением», чтобы иметь возможность делать выбор и использовать открытое ПО, в том числе и операционную систему со «страшным» названием Linux.

Базовый дистрибутив, который используется в офисе – ALT Linux, хотя я не считаю это оптимальным решением. К его преимуществам можно отнести: хорошую русификацию, неплохой форум по поддержке и огромное количество дополнительных пакетов из репозитория Сизиф. Против этого дистрибутива говорит не самая лучшая база драйверов для сетевых карт, в том числе и беспроводных, и вообще не самое удачное решение сетевых настроек. Но для обычного пользователя система вполне подходит.

Самая сложная часть при переходе – привыкание к новой среде – проходила под чутким управлением Владимира Сергеевича. Главным козырем была реальная возможность показать, что открытые программы ничуть не сложнее в использовании привычных большинству Windows и *MS Office*. Всё уже есть: полная русификация приложений, простота в установке и использовании. Включил, вставил диск, и через полчаса компьютер полностью готов к работе. Не нужно никуда звонить, узнавать ключи, коды и пароли. Всё очень просто!

Финансовая выгода сейчас, бесспорно, основное преимущество свободного ПО в бизнесе. Но есть ещё и свобода выбора, возможность изменения программ. Например, заказав приложение на основе свободного ПО для своего бизнеса у одной компании, вы можете через год пойти усовершенствовать её в другую фирму, ведь исходный код программы находится у вас, а не спрятан «в сейфе» разработчика. Надёжность свободного ПО избавляет системных администраторов от необходимости каждый день бегать и объяснять пользователям, почему пропал Интернет или завис компьютер. Им не нужно проводить целые дни, охотясь за вирусами или восстанавливая систему. У компаний появляется возможность не держать в штате слабоквалифицированного человека на постоянной основе, а нанимать временно со стороны высококлассного специалиста, ведь его присутствие будет необходимо только в случае расширения сети или установки новых приложений. Это, безусловно, серьёзная экономия средств фирмы и её времени, которое раньше тратилось на устранение мелких неполадок.

Почти все вопросы, которые возникали, решались достаточно быстро. Чаще всего решение помогали найти ссылки в Интернете на форумах. Использовать Linux для реальной работы – это просто, интересно и выгодно, что мы и демонстрируем на своём примере. LXF

SUSE: снова лидер?

Ну вот, погулял немножко – и вернулся. Выстоит ли SUSE 11 в битве с Ubuntu за наш настольный ПК?

Linux-разработка идет так стремительно, что мы в Башнях LXF едва успеваем за ней уследить. Возьмите хоть темпы выпуска: Ubuntu появился через десять лет после первой версии SUSE и за каких-то четыре года обскакал всех.

Рынок Linux-дистрибутивов ныне буквально набит новыми предложениями, и вы наверняка подумали, что SUSE для завлечения пользователей целиком сменил обличье. А вот и нет: SUSE не изменяет девизу «медленно, но верно». Ради этого даже иногда ломается шестимесячный цикл разработки (для многих дистрибутивов – табу). Перед нами SUSE Linux 11.0 – последняя версия одного из дистрибутивов-долгожителей: он вновь намерен вернуть себе место на Олимпе.

Взлёты и падения

SUSE – третий по возрасту среди нынешних Linux-дистрибутивов – ведёт родословную от Slackware, локализованного по-немецки. Несколько лет прочного соседства коробок SUSE и RedHat на полках магазинов принесло компании солидные прибыли. В поход на рынок корпоративного ПО в 2000 году оба дистрибутива тоже отправились вместе. Но подлинный переворот в SUSE произошел в 2004, когда компанию приобрела Novell: внезапно пошло массивное финансовое вливание, фокус сместился на предприятия, а приход коллектива программистов Ximian (предыдущего приобретения Novell) освежил кровь.

› **Ubuntu: для любителей коричневого. А также оранжевого.**

Увы, произошедшее поставило SUSE на грань исчезновения: после невероятного успеха версии SUSE 9.3 всё рухнуло под откос. Novell попыталась собрать все горошки на одну ложку – обновитель ПО *Yast Online Updater (YOU)* вылетел из дистрибутива, зато в нём возникла странная смесь из *RPM, Libzypp, Red Carpet* из *Ximian*, собственного *Yast* и Novell'овского *ZenWorks Management Daemon*. Казалось бы, при таком выборе найдётся всё и для всех. Но на деле получилась великая путаница, и многие просто ушли на совершенно другой менеджер пакетов, Smart.

Ubuntu устанавливал убогие темы и надоедливые звуковые эффекты в качестве стандартных, Mandriva не постеснялась включить рекламу в инсталлятор – и ничего, стерпели. Но негодный менеджер пакетов – это уже перебор. В SUSE 10.2 Novell постаралась исправить серьёзные ляпы, но на беду эта версия вышла спустя месяц после подписания патентного соглашения с Microsoft и утонула в волне презрительных (мягко выражаясь) высказываний, поднявшейся во всемирной паутине.

> **«Сделку с Microsoft люди восприняли как нож в спину.»**

Прыжок на подножку поезда

Природа свободного ПО такова, что большинство дистрибутивов использует одни и те же программы – ядро Linux, *X Window System*, *OpenOffice.org* и т.д. Конечно, где-то стоит ядро 2.6.26, а не .25, кто-то уже перешёл на *OOo* 2.4, а кто-то остался на 2.3, по-крупному расходятся Gnome и KDE. Но, как мы видим в Fedora и Ubuntu, можно поддерживать и Gnome, и KDE, и Xfce одновременно, а значит, различия между дистрибутивами постепенно стираются, и переключиться с одного на другой – проще некуда. В таких условиях один неверный ход может оттолкнуть массу пользователей; именно это и произошло после сделки с Microsoft.

Неудача с менеджером пакетов – крупная ошибка, но даже она не нанесла столь серьёзного ущерба репутации SUSE, как «братание» Novell с Microsoft. Не для корпоративных клиентов, нет – деловые люди всегда приветствовали сближение Linux и Редмонда. Но индивидуальные пользователи и энтузиасты восприняли эту сделку как нож в спину: возникло «народное» движение за бойкот Novell. Досталось даже *Linux Format* – после каждой публикации обзора SUSE или учебника по его программам (скажем, Tomboy) на нас обрушивался шквал электронных писем от читателей, осуждающих поддержку отщепенцев.

Хотя споры несколько поутихли, подспудная неприязнь к SUSE и другим проектам Novell, включающим придуманные в Microsoft технологии, остаётся. Но мрачные пророчества не сбылись: в SUSE ничуть не больше патентных ограничений, чем в любом другом крупном дистрибутиве, он по-прежнему использует то же ПО, что и прочие, и продолжает публиковать свои приложения под свободными лицензиями.

Люди не то что «простили» Novell, но склонны дать SUSE шанс – тем более, пробовать в новой версии таки есть что.

Готовый к установке SUSE 11 вы найдёте на нашем DVD. Но прежде позвольте упомянуть о наиболее заметных достоинствах и новшествах одиннадцатой версии…

› **Когда Novell подписывала сделку с Microsoft, Ричард Столлмен, наверное, рыдал.**

Любовь к KDE

Одно из привлекательных (для многих) качеств SUSE – его многолетняя преданность KDE. И Fedora, и Ubuntu, конечно, тоже предоставляют KDE, но только как проект второго плана. SUSE же вкладывает много труда в поддержку обоих рабочих столов. Полная версия DVD SUSE 11 (к сожалению, на наш диск она не влезла: HotPicks и прочий контент тоже немало «весят») предлагает выбор между Gnome 2.22, KDE 3.5 и KDE 4, и все они работают одинаково хорошо. Версия SUSE на нашем DVD включает KDE 4, а значит, вы сможете испытать новейшее ПО и увидеть SUSE 11 во всём блеске.

Бесспорно, KDE 4 пока не достиг диапазона функциональности KDE 3.5, но ведь он по большей части написан «с нуля», на новых технологиях типа Plasma и Phonon. Во всяком случае, SUSE 11 – лучший испытательный полигон для него: команда изрядно потрудилась, чтобы избавить «четвёрку» от тормозов и ошибок.

Qt 4 – вперёд!

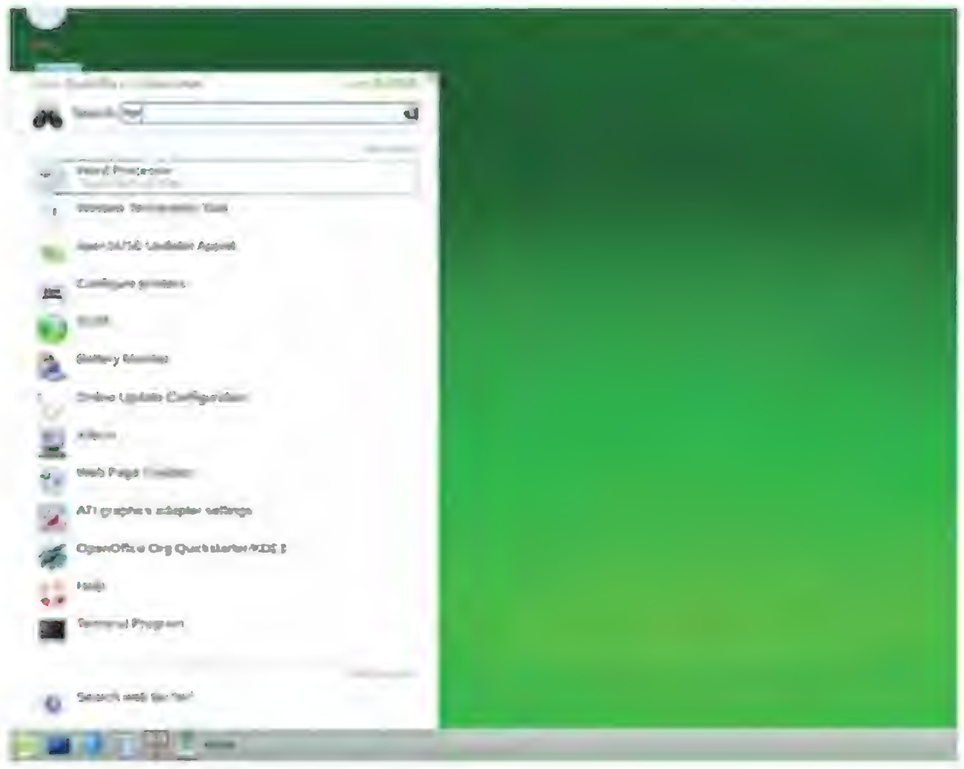
Приход KDE 4 означает, что на заднем плане маячит *Qt 4*. Одна из малоизвестных функций новейших версий *Qt* – возможность задавать стилевую информацию для виджетов пользовательского интерфейса (кнопок, флажков и пр.). Мы, конечно, знакомы с темами в KDE, но создавать их на удивление сложно: чтобы смастерить нечто неординарное, требуются солидные технологические знания. А вот таблицы стилей *Qt*, наоборот, сродни каскадным таблицам (CSS) для web-страниц – это текстовые файлы, и в них можно задавать цвета, размеры, изображения и прочее настолько просто, что теперь программисты с лёгким сердцем могут препоручить дизайн художникам.

Главный инсталлятор получил долгожданное обновление с использованием этих новых возможностей, и теперь он выглядит превосходно. Да, SUSE Live CD удобнее для пробы, но так вы упустите из виду новый дизайн! За несколько последних версий инсталлятор постепенно улучшился, и теперь разобраться в нём под силу любому новичку. Но и опытные пользователи не забыты: все продвинутые опции – вот они, под рукой. Нам понравился усовершенствованный менеджер разделов, с помощью которого минимально «подкованный» пользователь легко настроит LVM и шифрование. »

Взгляд изнутри: Нат Фридман

Технический директор Novell, отвечающий за стратегию в области Open Source

«Команда OpenSUSE отлично потрудилась над версией 11.0. Производительность системы управления пакетами *Zypper* значительно возросла, и на сегодня это лучший релиз OpenSUSE.»



› Меню приложений KDE весьма обширно, но его легко отфильтровать, введя название программы в текстовом поле вверху.

› Новый инсталлятор SUSE посрамил прочие установщики – не зря хвалили!

» Плазмоиды, за работу!

Разработка KDE 4 потребовала нескольких лет напряжённого труда, и теперь здесь есть на что посмотреть, даже в первые мгновения после запуска. Но, переварив первое впечатление от ярких новых значков Oxygen и общих визуальных изменений, вы поймёте, что наиболее заметное новшество для повседневной работы – система плазмоидов. Эти утилитки, управление которыми сокрыто в правом верхнем углу рабочего стола, впервые появились в *SuperKaramba*; но на сей раз это не только часики и метеосводки.

Отличие плазмоидов от виджетов *SuperKaramba* в том, что они не сбоку припёка, а встроены в самое ядро KDE. Kicker? По сути, плазмоид. Значки на рабочем столе? Тоже плазмоиды. Кому нужны старые виджеты *SuperKaramba* – пожалуйста, они работают в Plasma без всякой настройки. Идёт работа даже над интеграцией в KDE виджетов из Apple Dashboard (правда, о результатах говорить пока рано).

Возврат к KDE 3.5

KDE 4 не для всех – во всяком случае, пока. Многие из наших знакомых – поклонников KDE, поработав в новом рабочем столе пару недель, забросили его и вернулись обратно к KDE 3.5: слишком многого не хватало. Ещё неприятнее то, что различные ключевые программы, вроде *Amarok* и *KMail*, не имеют версий для KDE 4, и для их работы приходится загружать все библиотеки KDE 3.5 и *Qt* 3: дикое разбазаривание ресурсов! Другая проблема – четвёртая версия пока «не созрела», в отличие от 3.5, и возможны случайные зависания, странное поведение тем и нехватка некоторых функций.

Но SUSE сохраняет поддержку KDE 3.5 с использованием библиотек обратной совместимости (если хочется испробовать новейшее ПО и вы не против установки патчей). Предусмотрен и полный откат, со сносом четвёртой версии. Если вы раздобудете полный DVD с SUSE 11, то сможете выбирать между KDE 3.5, KDE 4 и Gnome 2.22 уже на стадии инсталляции.

> «Теперь программисты могут препоручить дизайн художникам.»

› В SUSE 11 по умолчанию предоставляется широкий выбор плазмоидов, и переделать рабочий стол – раз плюнуть.

Взгляд изнутри: Роджер Леви

Первый вице-президент и главный менеджер по открытым платформам в Novell

«Проект OpenSUSE – важная часть Linux-стратегии Novell, один из способов демонстрации нашей преданности сообществу свободного ПО. OpenSUSE служит основой для наших корпоративных продуктов и даёт Novell важное преимущество тесного сотрудничества с сообществом.»

В борьбе за совершенство

Яркие эффекты, переделанный *Yast* и – ух ты! – интероперабельность!

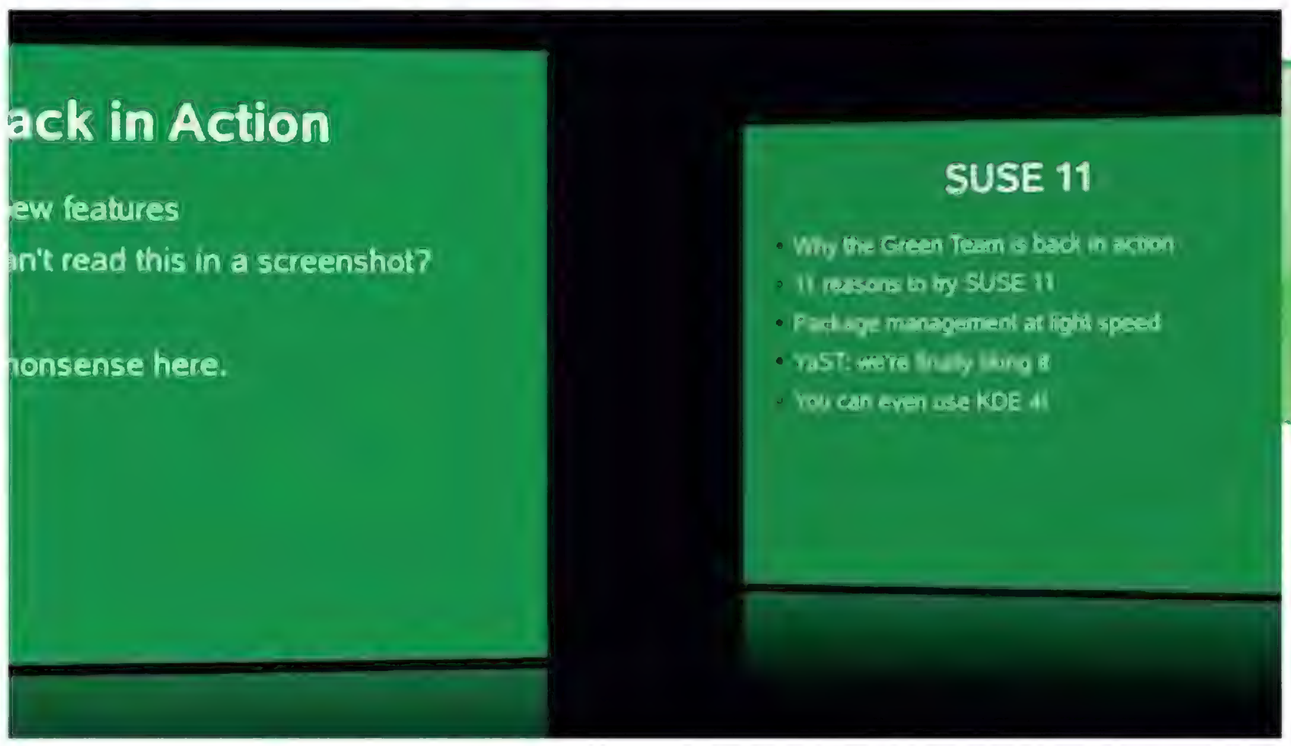
Принимая во внимание многочисленность программистов, работающих над открытыми проектами в Novell, неудивительно, что SUSE нередко во многом совершенствует базовые приложения, вроде *Firefox*, KDE и *OpenOffice.org*. Вот и сейчас, SUSE 11 первым представил абсолютно новую версию *OpenOffice.org* с поддержкой OpenGL.

Да, мы уже видели вращающиеся кубики и «желейные» окна, предоставленные *Compiz*, но все эти усилия обходили стороной прикладные программы: они продолжали оставаться плоскими. *OOo2.4* наконец ввёл объёмные эффекты на уровне приложения, а значит, теперь переходы между слайдами в *OOo Impress* могут поспорить с таковыми от Keynote на Mac'ах. Но 3D-эффекты требуют для работы и «правильной» 3D-карты, и «правильных» драйверов к ней. Результаты наших попыток с адаптером ATI и драйвером Radeon (текущая открытая версия) были настолько жалки, что нам пришлось взять проприетарный fglrx – только тогда всё наладилось.

Мы знаем немало людей, которым *Yast* нравился всегда, но в Башнях **LXF** таких не наблюдалось. Поймите нас правильно: *Yast* – очень мощная штука, выполняющая за вас немало работы по настройке системы. Только вот цена слишком высока: зачастую программа отрицает все изменения в файлах конфигурации, сделанные вручную. Но главным нашим нареканием была «неповоротливость» *Yast*: некоторые опции куда-то упрятаны, интерфейс наводнён сокращениями вроде NTP, NIS, LDAP, и даже твёрдо зная, что способ решения проблемы есть, его не всегда удавалось найти…

SUSE 11 вправе гордиться новым, переделанным *Yast* – он стал отличным инструментом администратора. Да, кое-где по-прежнему попадаются аббревиатуры, и на наш вкус, разобраться с программой чуть сложновато, но *Yast* сейчас в прекрасной форме. А новый *GTK*-инсталлятор? Он быстрее всех знакомых нам менеджеров пакетов, да и выглядит великолепно.

> «SUSE теперь вправе гордиться удобством инструмента Yast.»



› SUSE 11 одним из первых внедрил OOo с эффектами OpenGL для перехода между слайдами. Не бойтесь, на скорость загрузки это не влияет!

Вместе мы – сила!

Простая арифметика показывает, что Red Hat Linux имеет более 40 прямых «потомков» и несчётное количество косвенных (когда Distro A происходит от Red Hat, а Distro B – от Distro A). SUSE же пока сумел породить всего четыре. Mandrake, Conectiva, Caldera, Turbolinux, YellowDog, Centos, PCLinuxOS, Trustix, Lycoris, Foresight, RedFlag – всё это отпрыски Red Hat. А вот в SUSE настало затишье: после Astaro и неудачной Java Desktop System от Sun – ничего, хотя SUSE старше Red Hat!

Тому можно назвать две возможные причины, и обе связаны с *Yast*. Во-первых, прошло лишь четыре года с того момента, как SUSE присягнул свободе ПО, до этого же в лицензии *Yast* была строчка о том, что любое платное распространение его двоичного или исходного кода требует согласия SUSE. Во-вторых, *Yast* был задуман как центр управления дистрибутивом, и разработчикам было сложно вычленить какую-либо его часть.

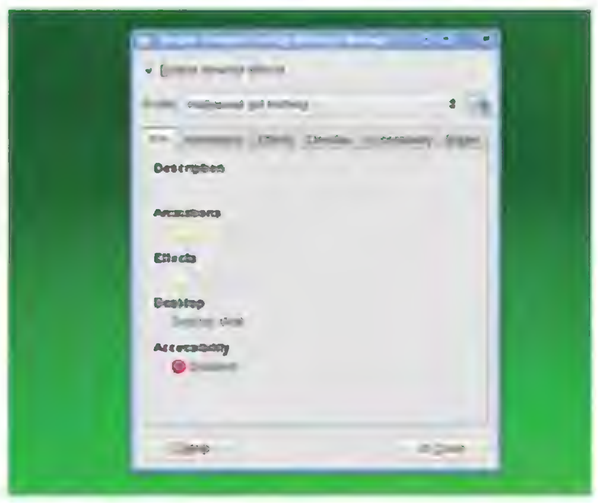
Перемены налицо: SUSE 11 поддерживает *PackageKit*. Если вы про него не слышали, не смущайтесь: *PackageKit* сравнительно юн, и SUSE внедрил его одним из первых. *PackageKit* – открытая среда для управления пакетами в любом дистрибутиве. Унификация стремится не к единому графическому интерфейсу, а к коллекции стандартных процедур, вызываемых кем угодно для установки программ и проверки доступных возможностей. Поддержка в SUSE *PackageKit* означает, что для управления ПО можно использовать как *Yast*, так и любой другой менеджер пакетов, на свой вкус (лишь бы поддерживал *PackageKit*).

А это открывает массу возможностей: *PackageKit* может устанавливать обновления безопасности во время загрузки системы, добавлять программы по требованию (т.е. в нужный момент) и даже позволять непривилегированным пользователям инсталлировать ПО в домашнюю директорию! Вдобавок консольный менеджер пакетов *Zipper* в SUSE 11 стал невероятно быстр, и команда разработчиков даже работает над портированием его в Fedora, как вариант для замены *Yum*.

Вероятно, сообщество отметит тот факт, что *AIGLX* теперь включается по умолчанию, в пику собственному детищу Novell, *Xgl*. Было время, когда расхождение трёхмерных рабочих столов казалось неминуемым, но SUSE 11 положил этому конец: зелёную улицу получил *AIGLX*.

Compiz — это просто

SUSE 11 включает *Simple Compiz Config Settings Manager* (для командной строки – *simple-ccsm*), позволяющий включить и настроить различные эффекты с помощью простого GUI. А если вы хотите просто активировать *Compiz*, выберите из списка профиль 'Hollywood Got Nothing'.



› Графические эффекты любят все! Включайте и наслаждайтесь.

В чем SUSE опережает

Если на 90% дистрибутивы одинаковы, мелочи подчас решают всё...

» Что бы ни говорили о Novell, нельзя отрицать, что здесь держат целую армию разработчиков свободного ПО. На все, от ядра и до KDE, найдется штатный сотрудник Novell, помогающий добавлять функции и исправлять ошибки. Один из побочных эффектов этого – выпуск основных версий многих программ «привязывается» к циклу SUSE; и отнюдь не потому, что разработчики придерживают их ради пиара для Novell. Наоборот, они работают сверхурочно, чтобы успеть с новой версией к моменту релиза.

Изюминка SUSE 11 – версия *Banshee 1.0*, и перечень усовершенствований весьма значителен. Вот наши любимые:

› **Простая и быстрая работа с плей-листом** Просто выбираете композицию и нажимаете клавишу Q: всё, нужный пункт добавлен в ad *hoc*-список!

› **Воспроизведение видео** Теперь можно сохранять и просматривать видео прямо в *Banshee*, как в *iTunes*.

› **Поддержка различных MP3-плееров** *Banshee* может даже конвертировать файлы в тот формат, который поддерживает ваш MP3-плеер.

› **Интеграция с *Brazero* для прожига CD**.

К тому же *Banshee* теперь быстрее работает и использует меньше системных ресурсов – всех победил!

Инсталляция одним щелчком

Нам очень нравится система установки программ одним щелчком, появившаяся в SUSE 10.3. Кто не пробовал, вот вам причина N 1. Обычно управление пакетами состоит из ввода URL репозитория ПО, просмотра длинного списка программ и выбора нужного пункта. Система работает неплохо, так как список репозиториев поставляется вместе с дистрибутивом, и вы своевременно получаете обновления. С другой стороны, вещи вроде AutoPackage позволяют загружать отдельные программы и быстро и легко устанавливать их; но менеджер пакетов не распознаёт такие загрузки, а значит, обновления патчами сильно усложняются. SUSE 10.3 предложил альтернативный вариант: так называемую установку одним щелчком.

«Система установки “одним щелчком” нам очень понравилась.»

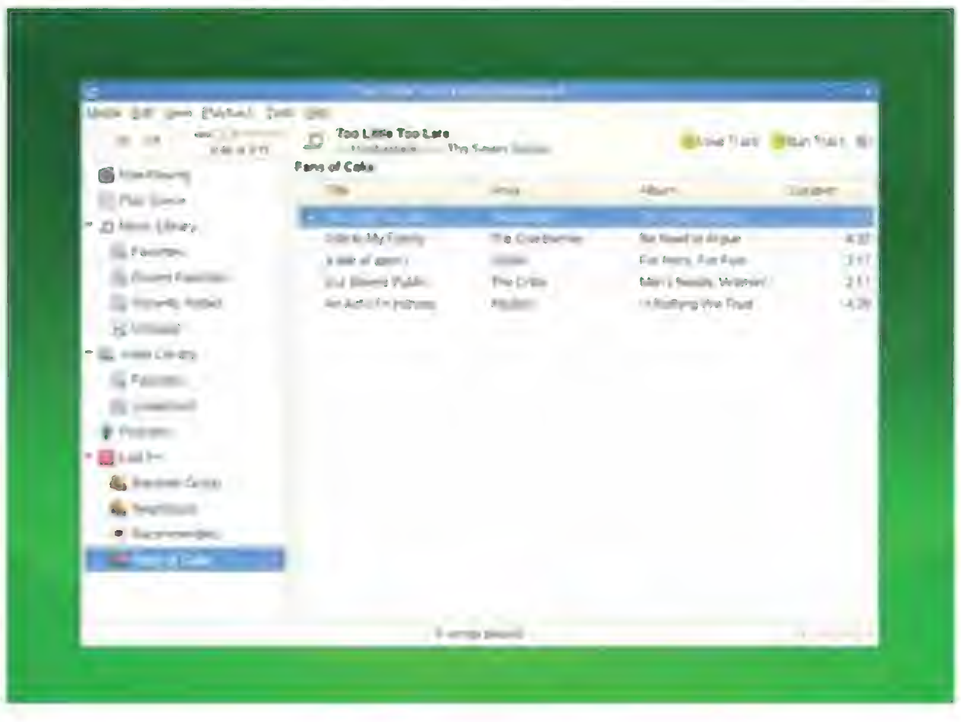
Эта система работает онлайн: вы замечаете окошко с предложением щёлкнуть по нему для установки приложения и кликаете; после этого (спросив, понятно, пароль), SUSE добавляет репозиторий в *Yast* и выполняет немедленную инсталляцию. Таким образом, вы получаете все выгоды своевременного обновления, избегая мороки со вводом URL в диалоговом окне. Полный каталог пакетов SUSE для установки одним щелчком хранится на http://software.opensuse.org/search, но некоторые сайты сообщества, например, PackMan (http://packman.links2linux.org), тоже содержат немало «однощелчковых» пакетов. Короче, таких уже несколько тысяч. Осталось только щёлкнуть мышью и следовать немудрёным инструкциям.

Mono: почти 2.0

SUSE 11 по умолчанию включает первую бета Mono 2.0 (Mono v1.9). Нет-нет, это не значит, что SUSE поддерживает нестабильное ПО – команда Mono считает все свои версии стабильными (как и команда разработчиков ядра Linux).

Здесь впервые применено совместное использование generic-кода (это уменьшает число генерируемых инструкций, повышая производительность) и включен *Silverlight*-совместимый движок *Moonlight*. А ещё в v1.9 был впервые активирован по умолчанию режим C# 3.0 – то есть команда Mono не отстаёт от Microsoft!

› **В хранилище ПО OpenSUSE полно программ, готовых к установке – знай щёлкай!**



› ***Banshee 1.0* не только свяжет вас с Last.fm, но и учтёт ваши музыкальные пристрастия.**

Взгляд изнутри: Ги Люнарди

Ведущий менеджер SUSE Linux Enterprise Desktop

«Люди просто бредят обновлённым инсталлятором. Впервые уже с ходу становится ясно, что устанавливается самая современная ОС. Пакет OpenSUSE 11.0 блестяще демонстрирует достижения открытого ПО.»

Убедитесь сами...

Загрузитесь с нашего DVD и опробуйте OpenSUSE 11.0.

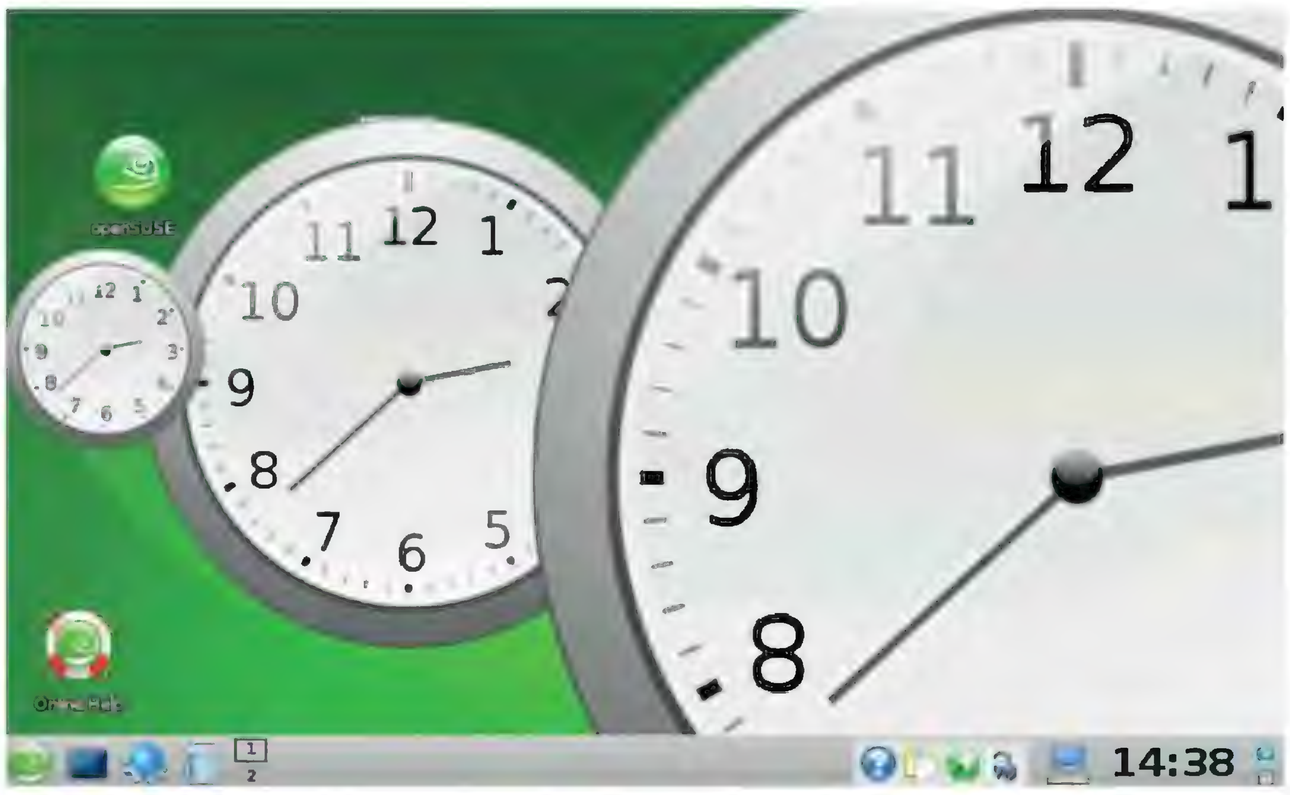
Пожалуй, хватит слов – пора оценивать самостоятельно. На DVD этого месяца есть Live-версия OpenSUSE 11.0, и её (вместе с KDE 4) можно испытать без установки на диск. Достаточно загрузить машину и посмотреть. А посмотреть есть на что:

» **Compiz** Как мы уже упоминали на стр. 43, настройка трехмерного рабочего стола через графический интерфейс значительно упростилась. К услугам опытных пользователей – командная строка и **compizconfig-settings-manager**, с ним можно добраться до любой мелочи.

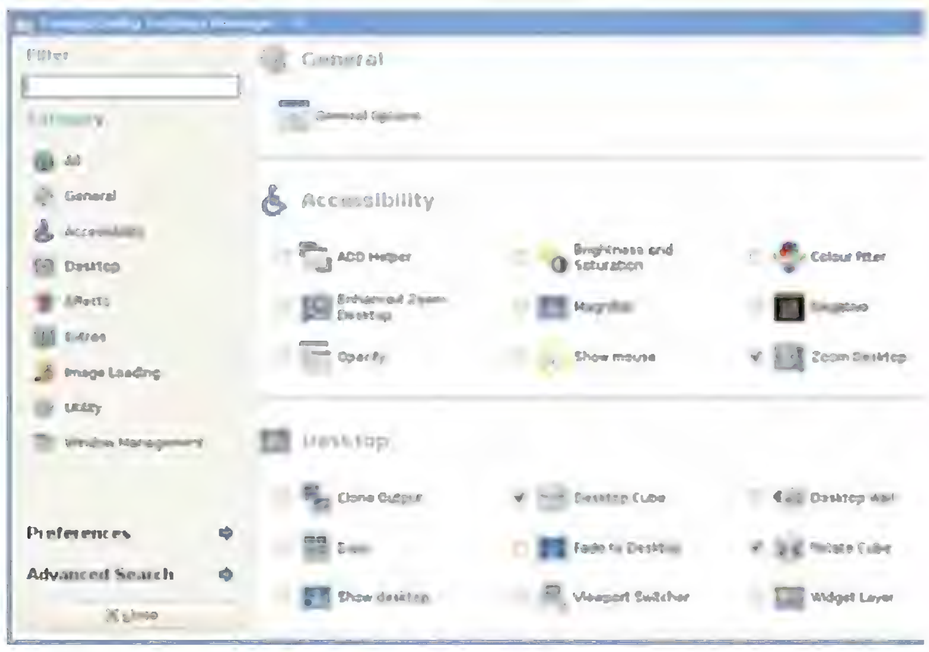
» **KDE 4** С KDE 3.5 тягаться рановато, но в исполнении OpenSUSE новый рабочий стол смотрится совсем неплохо. Щёлкните по жёлтому сектору в верхнем правом углу – и познакомитесь с племенем плазмоидов.

» **Yast** Его и хвалили, и хулили, но нам кажется, что инструмент администратора типа «всё-в-одном» достиг совершеннолетия. Если прежде вы отвергали *Yast*, взгляните ещё раз – он исправился и больше так не будет!

Итак, если вы довольны увиденным и хотите установить OpenSUSE 11.0 на жёсткий диск – перейдите на стр. 68 и почитайте полное руководство. А мы, как всегда, будем рады узнать ваше мнение о новой версии – не захотелось ли вам перейти на неё с другого дистрибутива? Черкните депешу на letters@linuxformat.ru! LXF



› **KDE 4 принёс целый короб модной косметики; пример – поддержка SVG (векторной графики): новые виджеты могут масштабироваться и не зависят от разрешения экрана.**



› **Любите *Compiz*? В этой утилите вы можете настраивать каждый пиксель любого эффекта рабочего стола.**

› **Ладно, заявляем официально: мы больше не ненавидим *Yast*. А если серьёзно, новый инсталлятор *Yast* и поддержка *PackageKit* возвращают SUSE в число бесспорных лидеров.**

Почему Linux готов войти в Большой Бизнес

Джеймс Литтон, бывший ИТ-директор Cray and Symantec America, считает, что Microsoft скоро приведет тысячи новых пользователей в Linux. Спасибо, ребята!

Наш эксперт

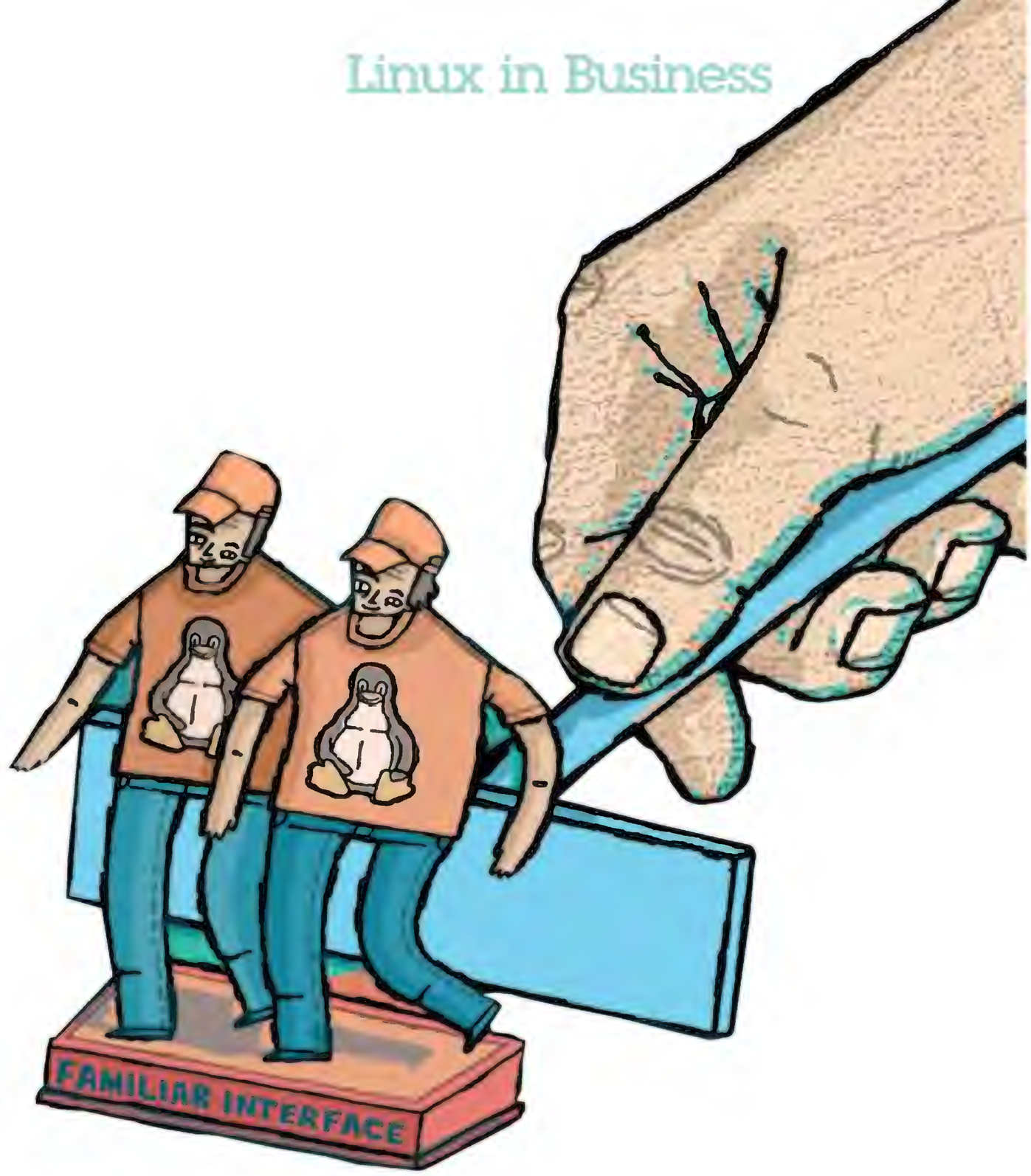
Джеймс Литтон занимал руководящие должности во многих организациях, ныне является партнером и исполнительным директором Identity Automation LP.

30 июня изготовители ПК прекратили продажи компьютеров с предустановленной Windows XP; 31 января 2009 Microsoft прекратит продажи Windows XP; а 14 апреля завершится централизованная поддержка Windows XP. Windows XP лидирует в мире по числу корпоративных установок, но ИТ-директоров, похоже, собираются вынудить перейти на новую операционную систему, невзирая на время и силы, потраченные на создание процессов и процедур поддержки вокруг XP.

> «ИТ-директоров, похоже, собираются вынудить перейти на новую ОС.»

Момент изъятия Microsoft'ом Windows XP с рынка все ближе и ближе, и пора проанализировать возможности, которые этот шаг открывает для перехода на Linux. Ценовое преимущество Linux для рабочего стола никогда не оспаривалось: ну что поставишь против бесплатности? Реальность в том, что для смены статус-кво ПО, «free» (имеется ли в виду свобода слова или бесплатное пиво) – недостаточно веский аргумент. Стоит лишь прикинуть, насколько глубоко Linux или иные свободные и открытые программы проникли на корпоративные ПК, чтобы, как это ни горько, осознать: «свободы» и «бесплатности» мало, чтобы ИТ-директора отказались от Windows или *MS Office* и заменили их на Linux или *OpenOffice.org*, при всех затратах на лицензирование и поддержку.

Так как же нам достичь нашей цели по увеличению количества свободных и открытых программ в коммерческой сфере? Последние несколько лет наблюдались значительные улучшения Linux в управлении пакетами, в инструментах GUI и в общей стабильности, и все это вместе взятое превращает его в конкурентоспособную систему для значительно большего числа пользователей, чем когда бы то ни было. Хотя пуристы не в восторге от этого, быстрое распространение инструментов GUI не только придает Linux практичности и гибкости, но и упрощает его сравнение с существующими решениями для настольного ПК. Я полагаю, что Linux – превосходное решение для любого уровня, но в корпоративном мире это не самое главное. Главное – найти решение, которое при желании можно



сравнить, по пунктам, с существующим (а в большинстве случаев это Windows) – и только тогда мы сможем убедить ИТ-директоров хотя бы задуматься о возможности применения Linux на рабочем столе.

Поскольку я занимал должности, предполагающие принятие решений в сфере ИТ, во многих крупных коммерческих компаниях, я могу утверждать, что технически наилучший продукт не всегда становится победителем, когда требуется соответствие внутренним потребностям фирмы. В процессе выработки решения нужно принимать во внимание множество факторов. Относительно настольных ПК, ведущими являются принятие пользователями и мощность, затем – способность ИТ-персонала управлять системой. Другие факторы, влияющие на этот процесс – стоимость, совместимость и/или наличие программ, совместимость форматов с существующими архивами файлов, а также кривая обучаемости и воздействие процесса на конечных пользователей.

Прижимистость казны

Если вы помните об успехе, достигнутом Linux в публичном секторе, вы, возможно, удивляетесь вышесказанному. Коли правительственные департаменты и образовательные учреждения стали переходить на Linux, то и все остальные тоже перейдут на него – это лишь вопрос времени, верно? Нет, неверно. Общественные организации и частный сектор подходят к затратам с совершенно разных позиций. Я вовсе не намерен огрублять картину принятия решений в государственных учреждениях, но нельзя отрицать, что организации, существующие на средства налогоплательщиков, куда больше зависят от стоимости, чем коммерческие фирмы, и в результате при противопоставлении решения, подразумевающего нулевые затраты на лицензии, и решения, подразумевающего хоть какие-то лицензионные выплаты, даровой вариант выглядит куда привлекательнее, даже если это новое решение не идет ни в какое сравнение со старым. Защищая Linux и пытаясь убедить ИТ-директора предпочесть его существующей системе, слишком многие из нас концентрируются исключительно на стоимости, и вот здесь-то наши усилия и пропадают впустую. Возможность сэкономить на цене зачастую кажется куда более интересной, чем все прочие оспариваемые возможности, но для нас главное – помнить, что финансовые затраты – это лишь часть дискуссии, но никак не вся дискуссия.

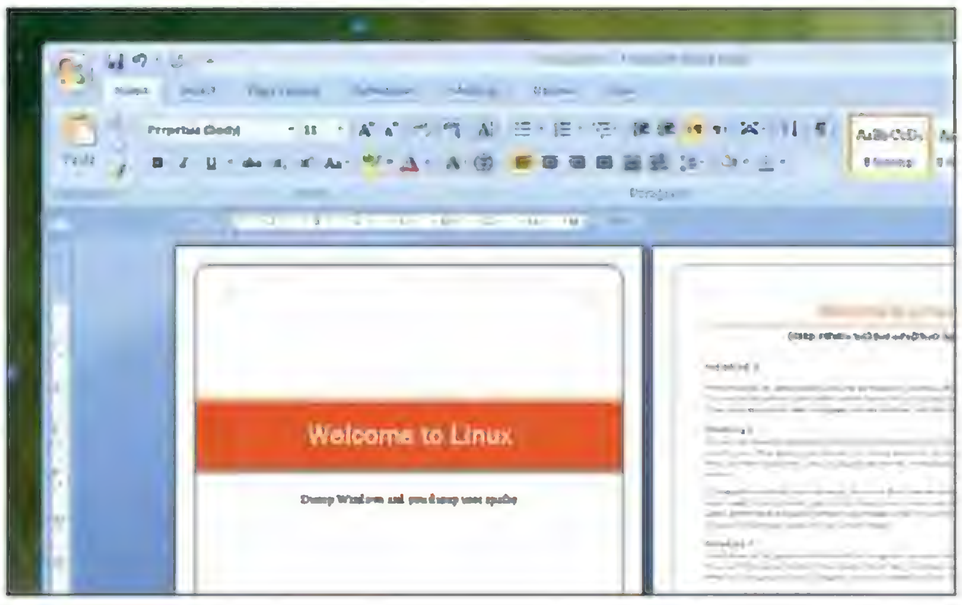
Хорошая новость для тех, кто отстаивает интересы свободного и открытого ПО, в том, что на данный момент мы имеем сильный, жизнеспособный и конкурентный набор инструментов – которого просто не существовало, или он существовал в весьма незрелом виде, еще несколько лет назад. Для примера возьмем *OpenOffice.org*. Раскладка и функции *OOo* сравнимы с существовавшими до 2007 года версиями *Microsoft Office*, а значит, переходя с *Office* на *OOo*, пользователи вряд ли заблудятся в новом интерфейсе. Еще одно преимущество *OpenOffice.org*, говорящее в его пользу – совместимость с форматами файлов *Microsoft Office*. И этот факт касается проблем совместимости, которые любой ИТ-директор непременно выдвинет как обязательное требование. С такими инструментами, как, например, *Evolution* для электронной почты, мы выполняем львиную долю требований по функциональности, характерных для большого процента средних корпоративных пользователей. А теперь добавьте сюда такие браузеры, как *Firefox*, *Konqueror* или *Epiphany* – и вот вам практически готовое решение. Но, к сожалению, остаются еще досадные обстоятельства, с которыми приходится считаться.

> «Главное — помнить, что финансовые затраты — только часть дискуссии.»

Динамический web

Одно из таких обстоятельств – сайты, основанные на проприетарных технологиях. Будучи сам владельцем бизнеса, я борюсь с этой проблемой. Я использую онлайн-версию *QuickBooks* для управления финансами, но, как вы можете догадаться, сайт требует *Internet Explorer*, поскольку он использует элементы *ActiveX*, и получается, что мне нужна физическая или виртуальная рабочая станция XP; а если я хочу работать в Linux, придется заставлять более старую версию *Internet Explorer* работать на моей Linux-машине, например, через технологию *Wine*. Любое из этих решений вполне жизнеспособно, но для специалиста по ИТ они оба работают на идею неготовности не-Microsoft'овского решения для «ответственного момента», побуждая покамест держаться того, что есть.

Еще одна реальная проблема – слабые наборы инструментов для таких областей, как управление проектами и создание диаграмм. Многие в корпоративном мире полагаются в своей повседневной работе на такие инструменты, как *Microsoft Project*. Хотя *Planner* и другие инструменты очень быстро заполняют этот пробел, нам все же предстоит еще пройти некий путь, чтобы сказать: да, у нас на платформе Linux есть все необходимые инструменты, доступные на других платформах; или – наши инструменты на 100% совместимы по функциональности с инструментами других платформ, например, Windows. При всем при том существуют разные сетевые инструменты, которые, буду-



› В *Microsoft Office 2007 MS* решил порвать с традицией и произвести резкие перемены в пользовательском интерфейсе.

чи соединены с ОС Linux и ее инструментами, дадут мириады вариантов замены дорогостоящих Microsoft'овских и чисто-Windows решений свободными или недорогими Linux- и web-приложениями.

Апология

Уверен, что многие из вас, читая эти строки, отнюдь не ликуют. Возможно, вы думаете, что свободное ПО не предназначено для эмуляции Microsoft, и не должно этого делать. Глубоко вдохните, и давайте еще раз рассмотрим нашу цель: она заключается в том, чтобы на рабочих станциях стало больше Linux и меньше всего остального. Это означает готовность выложить на стол сравнительный анализ, предусмотрев жизнеспособную альтернативу тем инструментам, которых на платформе Linux пока нет – или они есть, но им не хватает функциональности. Отсутствие родного решения в Linux для инструмента, имеющегося на платформе Microsoft, не означает, что мы должны развернуться и уйти, потерпев фиаско; просто надо взамен предъявить, например, web-инструмент, обеспечивающий такую же или похожую функциональность.

Сейчас важно, важнее чем когда-либо раньше, суметь проложить пути в корпоративную среду: ведь у нас действительно есть решение, способное поспорить с Microsoft практически на всех уровнях. И ключевой момент здесь – умение доказать это тем, кто принимает решения по приобретению технологий. Это означает, что помимо стоимости, мы должны выдвигать совершенно иные аргументы, и мы сможем это сделать, если сконцентрируемся на функциональности, управляемости и удобстве использования.

Еще одна проблема, с которой неизбежно столкнутся защитники свободного и открытого ПО – интерес ИТ-руководителей к поддержке и надежности тех программ, которые их просят рассмотреть. Большинство из тех, кто принимает решения в сфере ИТ, выбирает некое проприетарное ПО в предположении, что этот продукт соответствует их нуждам, но не менее важно и то, что им обещают: если что-то разладится, они получат адекватное содействие. И поэтому, наравне с демонстрацией сравнимой функциональности как аргумента в пользу свободного ПО, нам следует показать, что мы способны предложить не меньшую степень поддержки и надежности для открытых решений.

Для многих это болевая точка, и тем не менее, это крайне важно, и с этим необходимо считаться. Если вы, например, надеетесь убедить ИТ-директора отказаться от коммерческого решения типа Windows XP в пользу Linux, а не Vista, единственный ваш шанс преуспеть в этом деле – объяснить ему, что имеется более чем достаточная поддержка на случай проблем. Это означает коммерческую поддержку; и это означает, что деньги перейдут в другие руки. Пусть для некоторых это звучит ужасно, но это вовсе не порочит принципы свободного и открытого ПО – оно остается свободным и открытым, но вы платите за то, чтобы в нужный момент получить необходимую экспертизу. Это отнюдь не непреодолимое препятствие, и при правильном к нему отношении мы получим убедительный аргумент. ИТ-директора рассчитывают на выплату некой суммы за поддержку/эксплуатацию продукта – а в мире Open Source эти расходы намного меньше, чем в мире аналогичных проприетарных решений; и ИТ-руководители увидят функциональность, надежность и поддержку, сопоставимые с коммерческими решениями, но за меньшие деньги.

Принятие пользователями

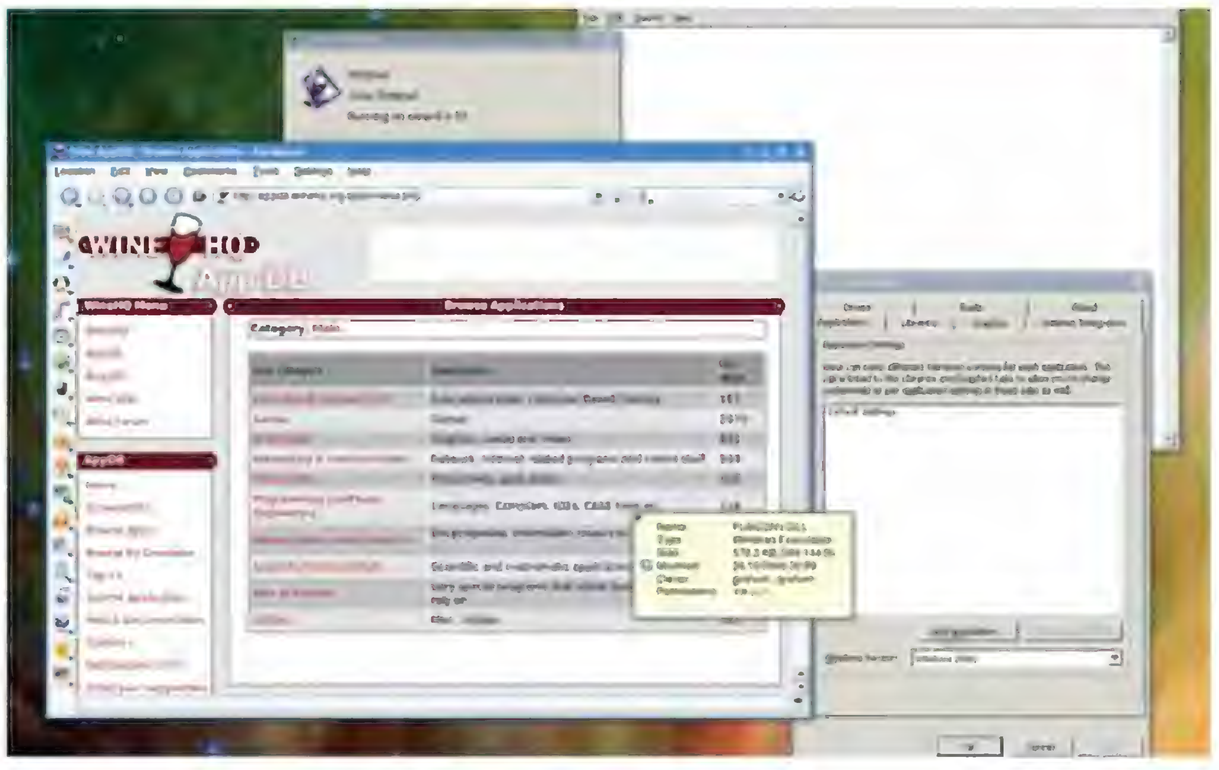
Если функциональность, надежность и поддержка лидируют в списке решающих факторов, то не менее важно и восприятие ИТ-директором удобства для пользователя. Мало что вселяет в сердце ИТ-директора больший ужас, чем предчувствие пользовательского бунта. Большинство предпочитает добиваться своего, прилагая максимум тонкости и дипломатии, лишь бы не связываться с пользователями. Мощное ПО наравне с мощной кампанией в пользовательском сообществе, пропагандирующей выгоды от предлагаемого перехода, почти всегда достигает успеха. Во многих организациях понадобится рассказ о финансовых преимуществах такого решения, или дискуссия в самом сообществе о свободных и открытых программах – все зависит от традиций организации и ее корпоративной культуры. Очень важно вникнуть в эти традиции и выйти к столу переговоров с домашними заготовками, снимающими с лидеров неизбежную тревогу о том, как сообщить о грядущих переменах коллективу.

Многие в сообществе FOSS с неприязнью относятся к крупным технологическим корпорациям, однако объективная критика Microsoft и

Wine: подходящий выбор

Для тех компаний, которые не могут найти родные альтернативы Linux для всех необходимых программ, *Wine* – отличная возможность заставить приложение Windows работать на платформе Linux без создания виртуальной машины. Такой подход может сократить затраты, причем новая среда покажется пользователям знакомой и привычной. Количество приложений, способных работать, используя библиотеки *Wine*, просто ошеломляет. Зайдите на **winehq.org** и посмотрите в AppDB – сами убедитесь.

Не огорчайтесь, если нужной вам программы нет в списке. Если у вас система, еще более запутанная, чем популярные приложения типа *MS Office*, вполне возможно, что вы будете первым, кто попробует запустить ее в Linux, и может оказаться, что после мелких переделок она прекрасно заработает.



› ***Wine*** **только что достиг версии 1.0 – важная веха в программной совместимости.**

других гигантов индустрии за их подход к обучению целой армии специалистов по поддержке их продукции далеко не гениальна. Нравится это кому или нет, но любой ИТ-руководитель, выбирая продукт, будет учитывать и этот фактор как решающий. Предложите отказаться от Windows в пользу Linux, и один из пяти первых вопросов коснется квалификации персонала, необходимой для поддержки новой платформы.

Обучение персонала

Как владелец собственного бизнеса, я постоянно ищу персонал с хорошими навыками работы в Linux, и реальность такова, что квалифицированных инженеров Linux значительно меньше, чем квалифицированных инженеров Windows. Самое подходящее решение этой проблемы – поэтапное внедрение при интенсивном обучении имеющегося персонала. Высоки шансы, что ИТ-персонал вполне способен обучиться поддержке настольного Linux.

Это не означает, что каждый сотрудник должен стать экспертом в Linux: им просто понадобятся базовые навыки для повседневной работы. Персонал следует укрепить экспертами, способными справиться с проблемами, ставящими в тупик команду поддержки первого, а возможно, что и второго уровня; понадобится всего несколько человек сверх вашего переобученного штата, и вы сможете сопровождать систему практически безболезненно. Как вы выполните данное требование – неважно, но надо учесть эту проблему и освободить от нее руководителя.

Начните прямо сегодня!

Сейчас позиции Linux, а также свободного и открытого ПО на платформе Linux, позволяющие им посягать на корпоративные ПК, сильнее, чем когда-либо прежде. Решение Microsoft прекратить поддержку XP и продвигать на новые ПК Vista означает, что у сторонников свободных программ появилась уникальная возможность склонить баланс от Vista к Linux в борьбе за место Windows XP.

> «Опережайте проблемы и вопросы, которые возникнут неизбежно.»

Ключ к успеху в том, чтобы мыслить не только в категориях экономии, но прийти к столу переговоров с готовыми планами, опередив все вопросы, неизбежные для типичного ИТ-руководителя. А это означает упор на функциональность, поддержку и дополнительное обучение ИТ-персонала компании, вовлечение сообщества, наличие эквивалентных или альтернативных инструментов для каждого приложения и достойное обеспечение поддержки и надежности продукта. Уделив этим вопросам должное внимание и эффективно представив их ИТ-руководителю, мы обнаружим, что на Linux перейдет еще больше коммерческих ПК во всем мире, и обеспечим Linux возможность успешно заместить XP. Теперь вы знаете, что делать – ну так идите и делайте! LXF

Windows на Linux?

Некоторым организациям волей-неволей придется по-прежнему держать критически важные приложения Windows, неспособные работать с инструментами эмуляции вроде *Wine* и не имеющие родных эквивалентов в Linux. К счастью, выход есть.

Один из вариантов для малых групп – взять небольшой инструмент виртуализации, например, *VirtualBox*; он позволит запускать Windows в Linux, обеспечивая, таким образом, приложению платформу для работы. Для программ, которые должны быть доступны большому числу пользователей, есть коммерческие решения, способные решить проблему, пока не появятся нужные приложения на основе web, или родные для Linux, позволяющие вашей организации отказаться от чисто Microsoft'овского решения.

› ***VirtualBox*** **сейчас разрабатывается в Sun Microsystems.**

Тайны сообщений об ошибках

Грэм Моррисон раскрывает тайный смысл наиболее частых сообщений об ошибках в Linux, и по мере изложения помогает исправить проблему.

Linux пугает некоторых уже тем, что выдаваемые им сообщения об ошибках, на первый взгляд, выглядят вестниками апокалипсиса. И их полно. Поискав на наших форумах слово «Error», вы получите более 150 страниц результатов. А это – много людей, столкнувшихся со многими проблемами.

Самая большая загвоздка для пользователей – не изобилие сообщений об ошибках; труднее всего что-нибудь по ним понять. Что, например, означает «Kernel Oops» или «PCI Can't Allocate»? Сообщения о сбоях Linux невнятны, сложны для понимания и редко когда полезны. И очень жаль, так как большинство проблем разрешимо довольно легко, причем значительное их число повторяется снова и снова. На языке бизнеса это низко висящий плод. Такие-то ошибки мы и рассмотрим.

Вовсе не нужно быть экспертом в Linux, чтобы машина успешно загрузилась, или программистом – для воспроизведения файла с фильмом. Но сообщения об ошибках все еще считают пользователя экспертом. Мы прольем свет на самые частые проблемы и дадим решения, которые помогут вам обойти их и вернуть свою машину в строй. Мы выбрали, на наш взгляд, наиболее проблематичные области: трудности с загрузкой, общие вопросы использования приложений, файловую систему, сеть и установку дистрибутивов.

Из каждой области мы взяли несколько наиболее частых ошибок, и, объясняя, что происходит, заодно предлагаем их решение. Наше намерение – дать вам, даже если эта проблема вас не касается, представление о том, как и почему сообщения об ошибках Linux кажутся загадочными и немного пугающими. И, надеемся, полученные знания о поиске лучшего решения помогут вам изгнать ваши собственные проблемы.

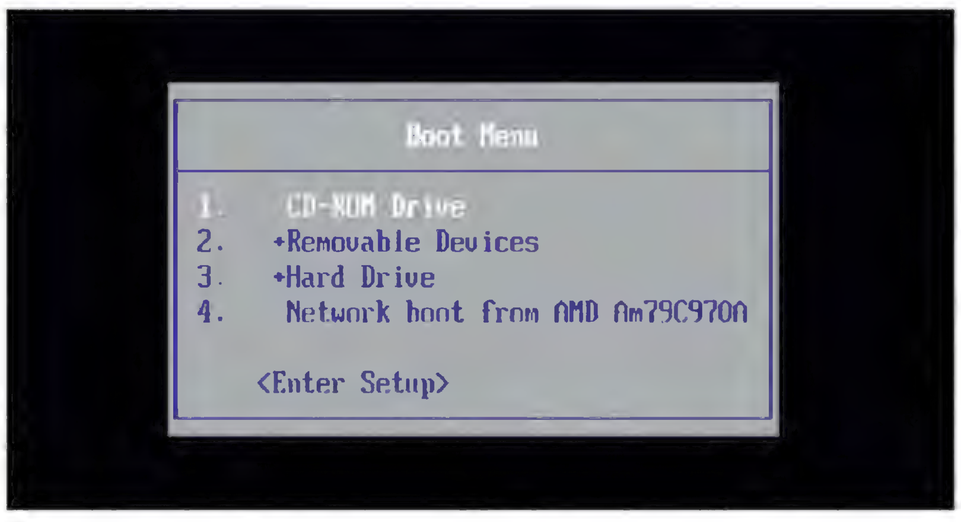
Установка дистрибутива

Любой дистрибутив Linux имеет собственную процедуру установки, и каждая вызывает проблемы. Ubuntu может работать на одной машине, а на другой – нет. Компьютер с Ubuntu может не дружить с Fedora, или OpenSUSE, или Linux Mint, или Mandriva…

ОШИБКА Невозможно загрузиться с CD/DVD

Часто для новичков в Linux это первое знакомство с ОС: вы вставили диск в привод и перезагрузились лишь ради того, чтобы вас поприветствовала ваша старая система. Проблема в том, что ваш жесткий диск имеет более высокий приоритет, чем оптический привод. Многие современные BIOS'ы содержат загрузочное меню, откуда можно изменить приоритет ваших устройств на лету – попробуйте нажать клавишу Escape или F12 при первом появлении чего-либо на экране. Оттуда легко выбрать загрузку с оптического диска.

У машин постарше может не быть такой функции. Тогда во время старта ПК вам придется нажать либо F2, либо клавишу Del, чтобы попасть в BIOS и поменять там порядок загрузки. Обычно эта опция сидит в меню «Boot», и вам следует сохранить изменения, чтобы загрузиться с оптического диска. Та же процедура имеет место при загрузке с внешнего привода или USB-брелка, что может пригодиться в интернет-кафе или за корпоративной машиной.



› Иногда для запуска Live-дистрибутива с оптического привода нужно поменять приоритет загрузочных устройств.

ОШИБКА PCI: cannot allocate

Существует много проблем, подобных этой, и большая их часть возникает во время загрузки. Причина тут одна: плохо работающее управление питанием. Виновник – некий ACPI, Advanced Configuration and Power Interface [Усовершенствованный интерфейс конфигурации и управления питанием]. Хотя это стандарт для управления питанием, он досаждает всем уже лет десять. Беда его в том, что драйверы устройств склонны не полностью поддерживать данную спецификацию. При каждом запуске управления питанием вашей машины, например, при загрузке или выходе из режима спячки, определенные устройства сбоят. Установка с Live CD усугубляет проблему: здесь нет такой роскоши, как проверка точного соответствия оборудования при загрузке, и тут не найдешь любой драйвер для любого устройства. Потому-то эта проблема и нередка при установке с Live CD.

Выход здесь только один: отключить ACPI. Иногда это можно сделать из вашего BIOS'а; а если нельзя, придется отключить ACPI во время загрузки. Вначале нажмите Escape, чтобы попасть в меню *Grub*; там выберите ту опцию, что вы обычно используете. Спуститесь к строке, начинающейся с **kernel**, и нажмите E, чтобы отредактировать ее. Добавьте в конец **acpi=off noapic** и нажмите Enter и B, чтобы начать процесс загрузки. Вы увидите, что машина запустится без проблем, и если вы продолжите установку Linux, теперь ваш дистрибутив лучше справится с подбором правильных драйверов.

Толкование сообщений об ошибках

Проблемы с загрузкой

Нет ошибок гнусней, чем те, что не дают вашей системе загрузиться – потому что нет инструмента решения проблемы. Увы, проблемы с загрузкой таки встречаются. Мы любим устанавливать дистрибутивы, и часто у нас их несколько на одной машине, да иногда еще и Windows на том же диске. Любая из этих установок может испортить процедуру загрузки, а заставить систему заработать снова не очень-то просто.

ОШИБКА Grub…

Если это все, что вы видите при включении машины, значит, повреждено загрузочное меню *Grub*. Это часть установки Linux, ответственная за загрузку ОС. Единственное, что здесь можно сделать – это запустить Linux с другого носителя, желательно с LiveCD. Дойдя до графического рабочего стола, откройте терминал, перейдите в режим root и введите **grub**. Кстати, это также пригодится, если пункты меню *Grub* не содержат разделов Linux.

Наберите **find /boot/grub/stage1**. Это поиск исходного загрузочного диска, и вы должны получить что-нибудь наподобие **(hd0,0)** (результат зависит от установки) – так в синтаксисе *Grub*'а именуются накопители. Теперь надо ввести **root (hd0,0)** (или ваш результат), чтобы сказать *Grub*'у, с какого раздела загружаться, затем **setup (hd0)** для переустановки загрузчика в главную загрузочную запись диска. Теперь после перезапуска компьютера вы сможете загрузиться нормально.

```
File  Edit  View  Terminal  Tabs  Help
       [ Minimal BASH-like line editing is supported.  For
         the  first  word,  TAB  lists  possible  command
         completions.  Anywhere else TAB lists the possible
         completions of a device/filename. ]

grub> find /boot/grub/stage1
 (hd0,0)

grub> root (hd0,0)

grub> setup (hd0)
 Checking if "/boot/grub/stage1" exists... yes
 Checking if "/boot/grub/stage2" exists... yes
 Checking if "/boot/grub/e2fs_stage1_5" exists... yes
 Running "embed /boot/grub/e2fs_stage1_5 (hd0)"...  17 sectors are embedded.
succeeded
 Running "install /boot/grub/stage1 (hd0) (hd0)1+17 p (hd0,0)/boot/grub/stage2
/boot/grub/menu.lst"... succeeded
Done.

grub>
```

› Умение переустановить загрузчик *Grub* на главный жесткий диск избавляет от многих проблем.

ОШИБКА Out of range
ОШИБКА Fatal server error: no screens found

Такое случается, когда предварительно настроенный экранный режим несовместим с вашим монитором. Нажмите Ctrl+Alt+F1 для перехода в консоль, войдите от имени администратора-root (или воспользуйтесь *sudo* из обычной учетной записи в Ubuntu). Пользователи дистрибутивов, основанных на Debian, для перенастройки экрана могут ввести **dpkg-reconfigure xserver-xorg**.

Остальным рекомендуем исправить свои настройки вручную, следующим образом. Во-первых, наберите **cd /etc/X11**, а затем **cp xorg.conf xorg.lxf**, чтобы сделать копию исходного файла конфигурации. Теперь откройте этот файл своим любимым редактором для командной строки. Кто не уверен, введите **nano xorg.conf**. Если вам известна спецификация вашего экрана, прокрутите файл вниз и найдите «Section Monitor». Здесь нужно вручную указать вертикальную и горизонтальную частоты развертки. Если вам неизвестно разрешение вашего экрана, прокрутите еще ниже и найдите раздел Screen. В нем нужно удалить все высокие разрешения экрана, так как мы ищем «наименьший общий знаменатель» (мы бы рекомендовали убрать все, что превосходит 1024x768). Когда графический рабочий стол будет починен, вы сможете увеличить разрешение. Если ни один из этих способов не помог, последний надежный вариант – поменять «Device» на "vesa", что полностью отключит ваш графический драйвер.

ОШИБКА Kernel panic!

Сообщение «Kernel panic» или «Kernel oops!» – максимальное приближение Linux к Синему Экрану Смерти, который все еще страшит сторонников Microsoft. Как и в случае Windows, при его появлении мало что можно предпринять, разве что отключить питание. Ядро – это сердце системы Linux, и Oops обычно вызывается дурным поведением оборудования, которое отсылает его в неразмеченную область системной памяти. Лучшим решением будет обновление ядра, так как проблема с оборудованием может быть решена в более новой версии. Но чтобы воспользоваться Linux для установки обновления, возможно, придется вернуться к более старой версии ядра в загрузочном меню.

Другой способ – распознавание оборудования-вредителя. Если вы сменили устройство или установили новый драйвер, скорее всего, причина в нем. Иначе вам придется поштучно изымать оборудование и смотреть, загрузилась ли машина. Несмотря на обильный вывод, понятной простому смертному информации сообщается очень мало.

ОШИБКА Неправильное имя пользователя или пароль

Вы удивитесь, узнав, сколько читателей звонят нам и говорят, что забыли свой пароль, или, хуже того, клянутся, что их о пароле и не спрашивали. К счастью, не все потеряно. Вам следует загрузить вашу машину в однопользовательском режиме или в режиме восстановления. Для этого нажмите на клавишу Escape, как только при запуске компьютера исчезнет экран BIOS'а. Вы увидите загрузочное меню *Grub*.

Если в этом меню нет ни пункта одного пользователя, ни режима восстановления, выберите то ядро, что обычно загружается (как правило, самое верхнее в списке), и нажмите E для редактирования параметров загрузки. Перейдите на строку, начинающуюся с **kernel**, и опять нажмите E для изменения ее текста. Убедитесь, что курсор стоит в конце строки, и добавьте следующее: **rw init=/bin/bash**. Нажмите Enter, а затем B, чтобы загрузиться. Это мы изменили опцию загрузки по умолчанию на вход в терминал *Bash*, а не в обычную сессию. Войдя в безопасный режим или режим восстановления, введите **passwd**, а затем имя пользователя, чей пароль вы хотите изменить. Без этого команда **passwd** поменяет пароль root'а. Потом перезагрузите компьютер и используйте уже новый пароль.

› Имея физический доступ к машине, можно легко поменять пароль пользователя, загрузившись в режиме восстановления.

Толкование сообщений об ошибках

Файловая система

Файловая система – это часть ОС Linux, ответственная за считывание и запись файлов, в том числе на внешние устройства. Как правило, она устойчива, но сбои питания и плохое оборудование могут иногда вызывать проблемы. А проблемы файловой системы обычно решать сложно. Для этого используется командная строка.

ОШИБКА Run fsck manually

Есть десятки разновидностей основной ошибки файловой системы. Такие проблемы обычно возникают при загрузке машины, и часто приводят к предупреждению о «Read only» корневой ФС. Это значит, что загрузка удалась, но делать вы ничего не можете. Решение – загрузиться с Live CD, так как при этом жесткий диск не используется и утилита восстановления сможет внести изменения, чтобы исправить проблему. Выполните команду **fsck -f /dev/drive**, заменив **drive** на устройство, соответствующее вашему корневому разделу: оно зависит от установки. Например, первый раздел на первом диске обычно называется **sda1**. Исходная ошибка должна содержать эту информацию. Запускать **fsck** надо от имени системного администратора; для пользователей Ubuntu это означает, что придется создать учетную запись с Ubuntu Live CD, введя **sudo passwd root**, а затем **sudo bash**.

ОШИБКА Устройство занято

Многие из нас используют USB-брелки и внешние жесткие диски, но иногда эти устройства отказываются отмонтироваться от файловой системы. Просто выдернуть их нельзя: тогда потеряются локально кэшированные и еще не записанные данные. Введя в командной строке **sync**, вы принудительно запишете на устройство все кэшированные данные, но устройство на волю не выйдет. Поможет команда **lsof** – не исключено, что ее придется установить отдельно. Набрав **lsof точка_монтирования**, вы получите список системных процессов, обращающихся к файлам устройства на текущий момент; и их следует завершить, чтобы система смогла отмонтировать носитель.

Это также пригодится, если у вас случатся неприятности с отмонтированием привода CD или DVD, поскольку техника выполнения та же (только здесь не нужно делать **sync**, ведь эти устройства только для чтения). Вот примерная сессия:

```
> umount /mnt/content
umount: /mnt/content: device is busy
> lsof /mnt/content
COMMAND PID USER FD TYPE DEVICE SIZE NODE NAME
smbd 23222 root cwd DIR 8,33 4096 2 /mnt/content
> kill -9 23222
> umount /mnt/content
```

Сеть

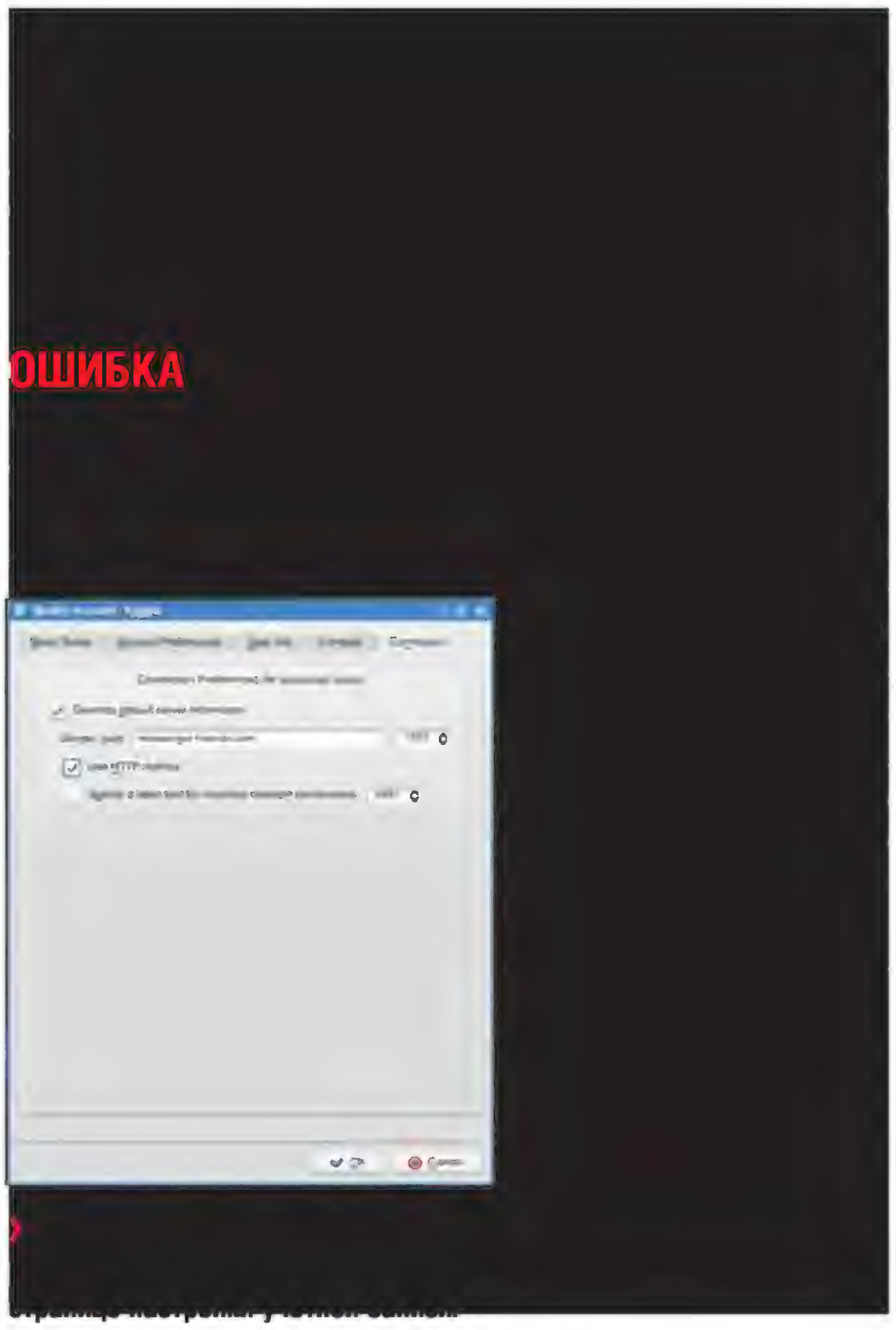
Любителей исправлять неполадки в сети на свете мало. Но в нашем мире, опутанном проводами, этого не избежать. К счастью, пара-тройка ошибок перекрывает большую часть проблем, и мы разрешим их для вас.

ОШИБКА Сервер не найден

Это классическая ошибка при работе с сетью. Включаете машину, ждете с минуту, пока она загрузится, щелкаете по ссылке на любимый сайт… и вместо него вас приветствует ошибка сервера. Проблема – в отсутствии соединения с Интернетом, и причин для этого может быть много. Лучший способ разрешить проблему – идти от главного соединения. Включен ли и работает ли ваш маршрутизатор? Работает ли на маршрутизаторе широкополосное соединение? Если у вас беспроводная сеть, следует, очевидно, проверить беспроводное соединение. В случае проводного Ethernet'а нужно проверить, горят ли оба светодиодных индикатора около кабеля: оранжевый сообщает, что соединение работает, а зеленый мерцает при любых действиях.

› Число причин этого сообщения об ошибке почти бесконечное.

Все это в порядке? Ищите проблему в Linux-компьютере. Проверив панель настройки сети вашего дистрибутива и убедившись, что все вроде бы работает как надо, попробуйте несколько утилит командной строки. **Ifconfig** дает не в меру обильный вывод, но это наискорей-



Толкование сообщений об ошибках

Приложения

В этой области применения Linux мы все испытываем затруднения. Пользователей OS X и Windows часто удивляет, как это сторонники Linux не могут просто скачать пакет из Интернета, дважды щелкнуть по нему и установить приложение без каких-то проблем. Они могут достать последние версии таких приложений, как *GIMP*, *Inkscape* и *OpenOffice.org*, просто скачав файл и запустив его. Пользователям Linux тут не повезло, и проблему усугубляет то, что многие дистрибутивы используют разные способы установки.

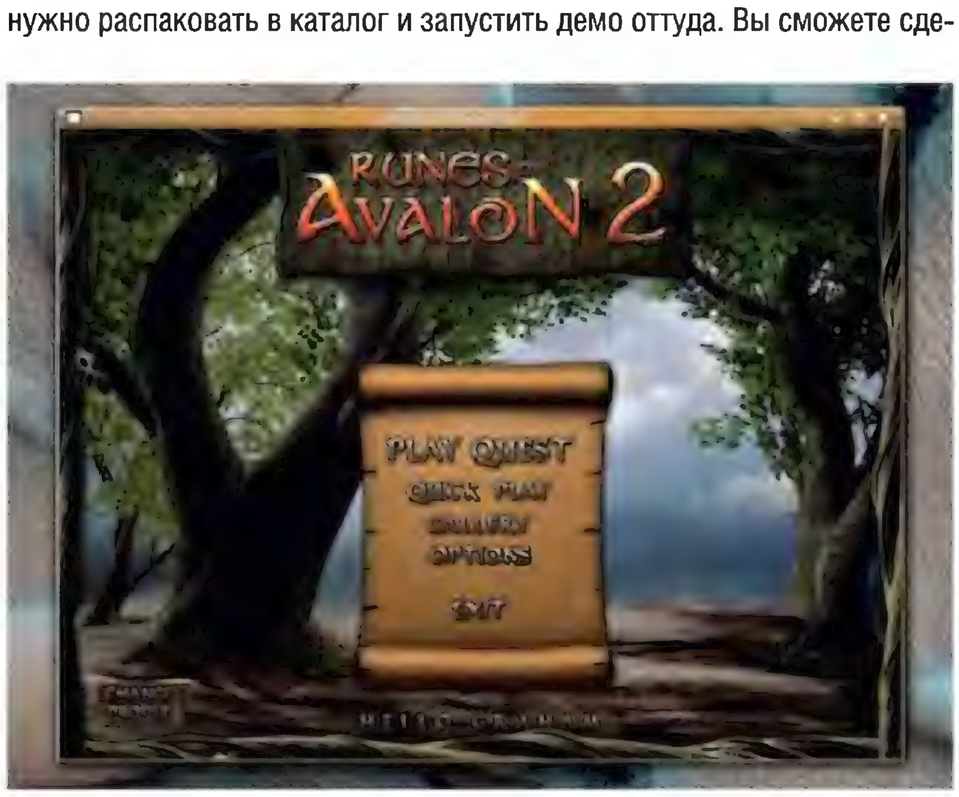
ОШИБКА Permission denied

Эта ошибка – защитная реакция системы, она часто возникает при попытке запустить приложения в командной строке и редактировать определенные файлы. Linux ограничивает доступ к некоторым файлам и каталогам, и даже если учетная запись пользователя взломана, злоумышленник не сможет испортить важные системные приложения. Такая система практичнее всего на сервере или на машине с сотней пользовательских записей. И хотя это важно и для однопользовательской машины, нет особой беды в том, чтобы немного отойти от предосторожности и дать себе права на запуск или открытие таких файлов.

Вы можете сделать это и из рабочего стола, и из командной строки, но потребуется использовать учетную запись администратора, чтобы иметь право менять требуемые разрешения. В командной строке систем типа Debian это означает ввод **sudo bash**, а в не-Debian – просто **su**. Можно поменять владельца файла, введя **chown имя_пользователя имя_файла**; ввод аргумента **-R** рекурсивно поменяет владельца файлов в каталоге. Но это не поможет другим пользователям вашей машины: они все еще будут испытывать проблемы с правами доступа. Ответ состоит в установке прав на запуск файла так, чтобы его могли выполнять все. Это достигается командой *chmod*. Введите **chmod +x имя_файла**, чтобы дать всем пользователям вашего компьютера права на запуск файла. Аналогично, **chmod +rw имя_файла** разрешит всем его выполнение и изменение. [Совет из серии «вредных» – защитные механизмы лучше не отключать, а разбираться, почему они вдруг сработали, – *прим. ред.*].

ОШИБКА Скачанное не запускается!

Как-то раз мы включили на наш DVD демо-версию одной игры, упаковав ее в файл **tar.gz**. Многие из нас, закаленных на полях Linux, этого даже в голову не взяли. Но нам позвонили несколько новых пользователей Linux (а их-то расположение нам и надо завоевывать!) и спросили, почему не запускается файл **tar.gz**. Ответ, конечно, в том, что **.tar.gz** – это архив. Мы объяснили, что он подобен файлу **zip**, то есть его нужно распаковать в каталог и запустить демо оттуда. Вы сможете сделать это почти во всех графических средах с помощью правого щелчка мыши, или же в командной строке, введя **tar xvf имя_файла.tar.gz** – ну, новичкам об этом знать не обязательно. Затем найдите файл **.bin** или **.sh** и щелкните по нему, тем самым запустив демо. Если вам опять не повезет, придется, видимо, набрать **./install.sh** из командной строки в только что созданном каталоге. От лица всех защитников Linux приносим извинения за неудобства.



› Для тех, кто не привык к Linux, запуск демо-версий не так прост, как в других операционных системах.

ОШИБКА Flash не проигрывается

Новые пользователи Linux, войдя в свежеустановленную систему, очень обижаются, что они не могут проводить круглые сутки на YouTube. Да, очень немногие дистрибутивы Linux включают по умолчанию поддержку воспроизведения Flash. Хуже того, безнадежно оптимистичное заявление браузера о том, что установка доступна за несколько щелчков мыши, оказывается, как правило, ложью. Однако есть надежный способ заставить работать поддержку Flash.

Поищите в Google «adobe flash download» и щелкните по первому результату поиска. На новой странице из выпадающего списка «version to download» выберите «tar.gz for linux» и щелкните по соглашению. Файл будет скачан, а вы хорошенько запомните место, куда браузер сохранит его. Затем откройте терминал и введите **cd** и пробел, после чего укажите путь каталога, где лежит загруженный файл (обычно это **~/Desktop** – ваш рабочий стол). Перебравшись туда, введите **tar xvf install_flash*** для распаковки скачанного пакета, и затем **cd** в новый каталог. Теперь можете запустить установщик, введя **./flashplayer-installer** и следуя экранным подсказкам. Браузер перезапустится, и вы увидите, что Adobe Flash работает. LXF

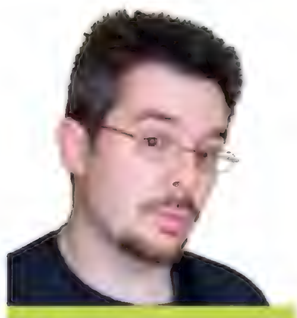
› Даже если поддержка Adobe Flash не входит в дистрибутив, ее можно поставить за две минуты.

Обратная связь

Разных ошибок на свете тысячи, а мы описали только с десяток. Но нам бы хотелось, чтобы с этого началось некоторое подобие FAQ, где мы могли бы помочь разобрать еще более загадочные сообщения. И нам нужна ваша помощь, чтобы узнать, какие сообщения, на ваш взгляд, самые неприятные и больше всего отталкивают от Linux новых пользователей, даже не давая им начать. Если у вас есть примеры невнятных ошибок Linux, пришлите их нам, мы объединим самые интересные и известим вас о результатах.

Свобода самовыражения

Дэниел Джеймс решил не просто писать о Linux, а заняться делом и помочь местной школе с информационными технологиями. Вот что у него вышло…



Наш эксперт

Дэниел Джеймс – один из сооснователей дистрибутива 64 Studio. Он обожает определять и устранять проблемы с ALSA по коротким и невразумительным описаниям.

В Великобритании государственные школы обязаны следовать национальному учебному плану, составленному управлениями центрального правительства. Современная методология для начальных школ, названная «Умение и удовольствие» ('Excellence and Enjoyment'), была опубликована в 2003 году. Этот документ расширил сделанный ранее акцент на обучении чтению, письму и счету, придав особое значение другим предметам, включая музыку. Web-сайт Министерства по делам детей, школы и семьи говорит об этой методологии: «Будет сделан акцент на искусстве и творчестве, и учителям для успешного самообучения и преподавания будет оказана помощь в использовании информационных технологий». Сегодня есть множество компьютерных программ для занятий творчеством, что позволяет достичь обеих целей сразу.

«Компьютерная грамотность» широко признается обязательной для школьников, так как дает им необходимый багаж для обучения и будущей работы. Но имеет ли это значение для «пост-широкополосного» поколения? Сейчас английские ученики поступают в начальные школы, уже имея опыт работы на компьютере, игровых консолях и прочих бытовых встраиваемых системах. К шести или семи годам многие дети являются опытными пользователями Web, проводя свободное время за сетевыми многопользовательскими играми вроде *Disney's Club Penguin*. Мало кого нужно учить пользоваться браузером или меню «Пуск».

> **«Свободное ПО дает школам ряд ценных преимуществ.»**

Если принять знакомство с компьютером как данность, то родителям и профессиональным педагогам, наверное, пора спросить, какие реальные выгоды должны давать информационные технологии в школе. Есть сетевые ресурсы для обучения грамоте и счету, которые почти полностью вытеснили образовательные CD-ROM, популярные на платформе Windows в девяностые. Но современные компьютеры умеют гораздо больше, чем воспроизводить готовый мультимедиаматериал; сегодня они – это рабочий инструмент во многих творческих областях деятельности, включая создание музыки, радиовещание и работу диджеев. Как школам научить детей пользоваться преимуществами данных технологий и извлечь большее из своих вложений в оборудование?

Что не так с ИТ в школе?

Одно из препятствий использования ПО для творчества в школе – высокая стоимость проприетарных лицензий. Если, по-вашему, *Microsoft Office* слишком дорог, полюбопытствуйте, сколько стоит *Creative Suite от* Adobe или *Nuendo* от Steinberg. Даже с учетом образовательных и «оптовых» скидок проприетарное творческое ПО стоит недешево, особенно если покупателю нужна полноценная версия без ограничения возможностей. Завучи и директора школ, не осведомленные о Linux, вероятно, считают, что такие программы им просто не по карману.

Если школа выбрала проприетарное ПО, нельзя ожидать, что у учеников будет доступ к этим приложениям дома. Образовательные учреждения не вправе поощрять использование пиратских продуктов, и в любом случае для работы многим программам нужны защитные USB-брелки с ключами.

Другое препятствие – недостаток детальных знаний о таком ПО у школьного персонала. Зарплата ИТ-лаборанта в начальной школе довольно низка, и едва ли эта должность соблазнит профессионала с большим опытом. Учителя без «творческого прошлого» тоже вряд ли обучались работе с такими программами. Наконец, взрослые час-

то недооценивают детские компьютерные способности, неумышленно ограничивая их возможности и здоровое честолюбие.

Свободное ПО, и в частности, Linux, дает школам ряд ценных преимуществ. Прежде всего, вероятно, бросается в глаза выгода от сокращения расходов на лицензирование, но есть и множество скрытых достоинств, осознаваемых лишь после развертывания новых систем. Для постоянных читателей это не будет сюрпризом, так как мы уже много писали о состоянии свободного ПО в образовании; но на сей раз мы собираемся вложить деньги (ну, то есть время и усилия) в нашу местную школу.

Начальная школа Всех Святых (All Saints) во Фрешуотере на острове Уайт – базовое образовательное учреждение с примерно 160 учениками от пяти до девяти лет. Через два года (или около того) она повысит уровень, начав принимать учеников до 11 лет. В прошлом году были выделены существенные средства на обновление компьютерного класса.

И мы создали музыкальный клуб!

В порядке добровольного содействия от родителей, я предложил установить программы на списанные компьютеры, чтобы создать музыкальный клуб, действующий после занятий или во время обеда. Приобретя новое оборудование, школа уже избавилась от большинства компьютеров предыдущего поколения. Я установил 32-битную версию дистрибутива Linux 64 Studio 2.0 на три старых машины: ПК с процессором Celeron 1 ГГц (бывший школьный файловый сервер), настольный ПК с Pentium III 500 МГц и Dell с процессором Pentium II 233 МГц. Первые два компьютера имели по 256 МБ оперативной памяти; самому медленному ее нарастили до 384 МБ (дополнительные модули на eBay стоили где-то пять фунтов). На компьютере с процессором Celeron был встроенный звук AC97, а Pentium III был оснащен звуковой картой M-Audio Audiophile 24/96 с MIDI-портом; на Dell был встроенный звук с чипом Yamaha OPL3 – обычный вариант тех времен, когда программный синтез аудио отнимал слишком много ресурсов центрального процессора.

Два более мощных компьютера получили рабочий стол Gnome дистрибутива 64 Studio, а на Dell был настроен *Fluxbox*, чтобы интерфейс не тормозил слишком сильно. Компьютер с Pentium II как мультимедийная рабочая станция, конечно, не лучший вариант экономии, но мне было любопытно посмотреть, на что способен древний ПК, выведенный в расход. У каждого компьютера был 17-дюймовый ЭЛТ-монитор, а такие сейчас считаются токсичными отходами – их нельзя даже выбросить. Требования по эргономике, принятые в самой школе, не разрешают детям пользоваться ЭЛТ-мониторами с меньшей диагональю.

All photos: Martha James

После небольшой тренировки дети могут играть аккорды на обычной Qwerty-клавиатуре.

В школе была одна клавиатура электронного пианино с MIDI-разъемом, авто-аккомпаниатор Yamaha, выдающий аккорд по нажатию одним пальцем. Ее приставили к делу как MIDI-контроллер для компьютера Pentium III с выходом, подключенным к мини-Hi-Fi. К двум другим компьютерам были подключены дешевые колонки, по типу игровых.

Как мы выбирали приложения

При выборе программ для музыкального клуба самыми важными факторами были легкость изучения и ясный, простой интерфейс. Дело не в том, что детям не справиться с более сложными программами типа *Ardour* или *Jamin* – просто они неусидчивы и теряют интерес, если нельзя сразу увидеть и услышать результаты.

Из синтезаторов любимцем детей сразу стал *AmSynth*. Почти весь его интерфейс размещен в одном окне, и хотя члены музыкального клуба никогда не брали в руки настоящий синтезатор, который эмулируется программой, было очень просто показать им, как перемещение регуляторов мышью изменяет звук.

Мы также установили *Alsa Modular Synth* – эмулятор модульного Moog-синтезатора шестидесятых, и *ZynAddSubFX* – другой очень продуманный программный синтезатор с почти бесконечным количеством настроек (есть даже настройка микротонов). К счастью для юных пользователей клуба, в нем есть пресеты, названные по эмулируемым »

На многих несложных электронных клавиатурах есть выход MIDI, что позволяет применять их как контроллеры на ПК.

AmSynth **– любимец детей в музыкальном клубе, хотя они никогда не видели настоящего синтезатора.**

инструментам. Интерфейс у *ZynAddSubFX* капризный и иногда может работать нестабильно, но он создает отличное звучание.

При запуске *ZynAddSubFX* предлагает два режима работы, включая режим для начинающих пользователей, с виртуальной MIDI-клавиатурой. Если у вас мало настоящих MIDI-инструментов, он весьма удобен: можно музицировать с помощью мыши или Qwerty-клавиатуры. В меню *Utils AmSynth* также есть виртуальное пианино (за приложения, не имеющих данной функции, эту работу делает *Vkeybd*). В случае с *Alsa Modular Synth* MIDI-соединение *Vkeybd* и выходов аудио со звуковым сервером Jack нужно выполнять вручную в окне Connections *QjackCtl* (Jack Control в меню Sound & Video 64 Studio).

Синтезатор *ZynAddSubFX* содержит массу настроек и звучит гораздо лучше, чем выглядит.

Байты и биг-биты

Эмулятор драм-машины *Hydrogen* тоже очень популярен у нас в клубе. С ним поставляется большой набор акустических и электронных ударных инструментов, и у большинства из них есть демо-шаблоны. Дети моментально схватывают идею редактора шаблонов, в котором прокручивающийся курсор отсчитывает время, а клики мышью добавляют или удаляют такты.

Для записи и редактирования звука была выбрана *Audacity* с подключенными напрямую к звуковой карте дешевыми конденсаторными микрофонами с питанием от батарейки типа тех, что используются с портативными видеокамерами. Так как *Audacity* не использует эффекты реального времени, аппаратные требования тут весьма скромны. Интерфейс *Audacity* понятен для детей, в частности, благодаря большим и четким кнопкам «Стоп», «Запись» и «Воспроизведение». При среднего качества аудиокарте, мини-микшере и конденсаторном микрофоне получившаяся запись по качеству может легко превзойти сделанную на кассетном магнитофоне.

«Большинству детей доступен компьютер, но никак не пианино.»

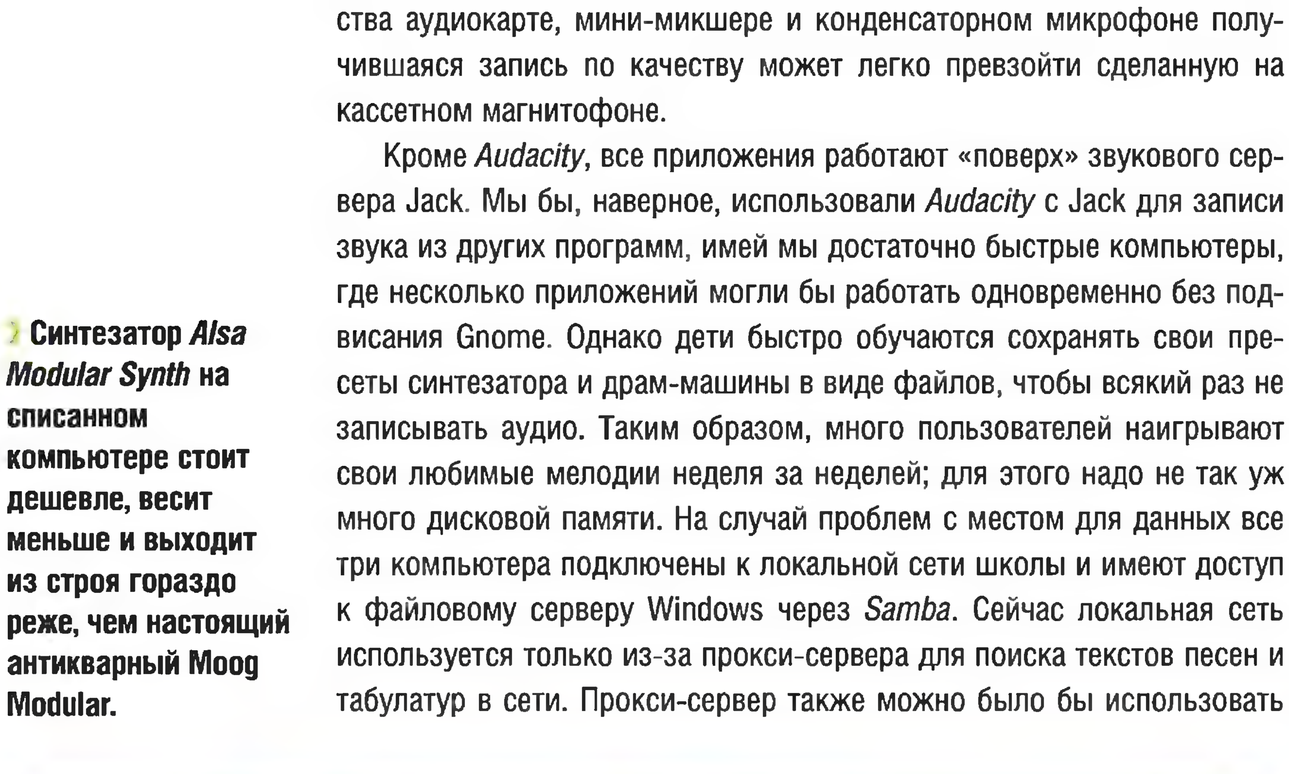
Кроме *Audacity*, все приложения работают «поверх» звукового сервера Jack. Мы бы, наверное, использовали *Audacity* с Jack для записи звука из других программ, имей мы достаточно быстрые компьютеры, где несколько приложений могли бы работать одновременно без подвисания Gnome. Однако дети быстро обучаются сохранять свои пресеты синтезатора и драм-машины в виде файлов, чтобы всякий раз не записывать аудио. Таким образом, много пользователей наигрывают свои любимые мелодии неделя за неделей; для этого надо не так уж много дисковой памяти. На случай проблем с местом для данных все три компьютера подключены к локальной сети школы и имеют доступ к файловому серверу Windows через *Samba*. Сейчас локальная сеть используется только из-за прокси-сервера для поиска текстов песен и табулатур в сети. Прокси-сервер также можно было бы использовать для загрузки пакетов обновления с *Synaptic* после выхода свежих релизов дистрибутивов.

Настоящие инструменты

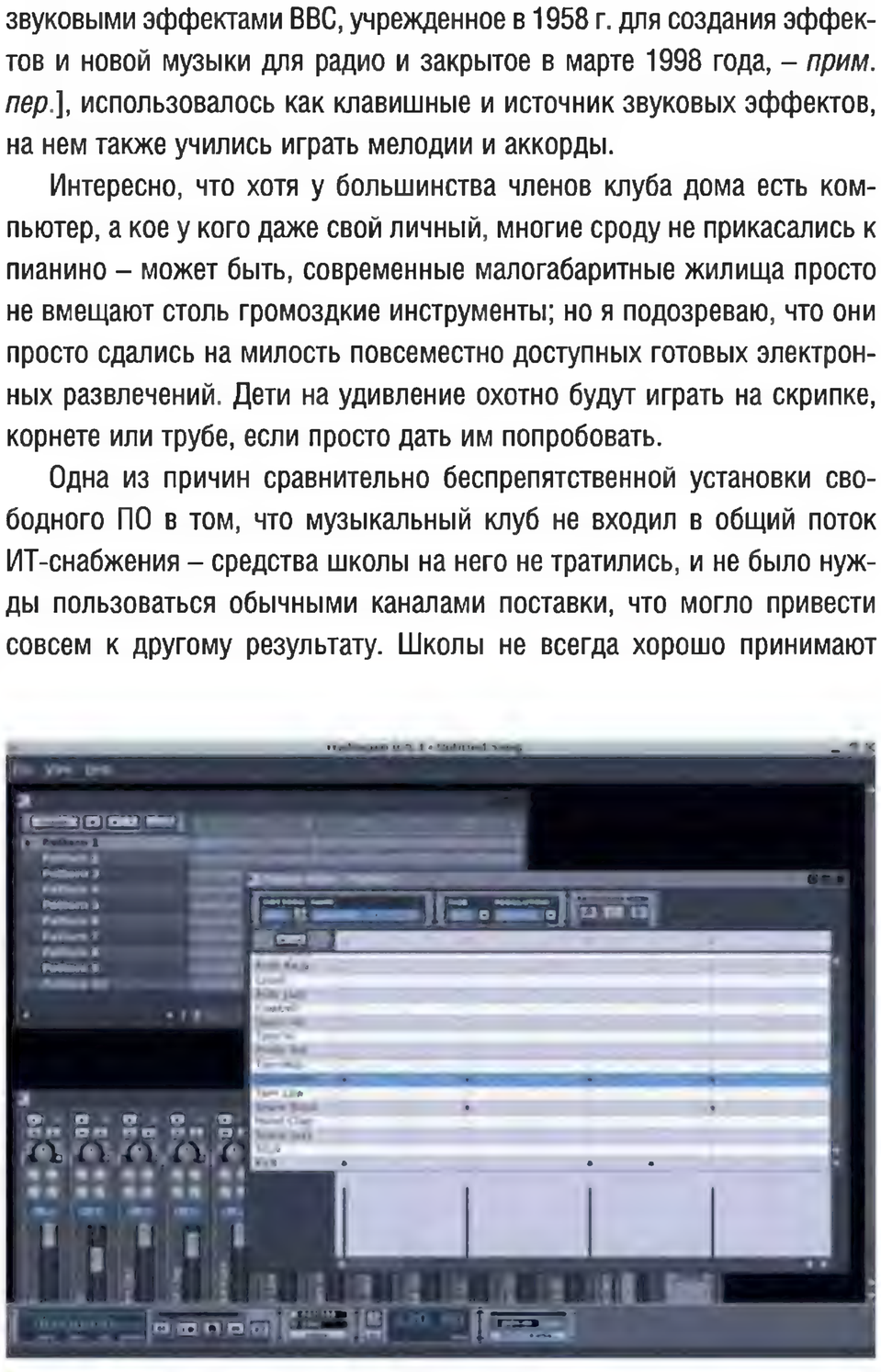
Микрофоны позволяют детям сочетать компьютер с обычными инструментами. Не считая ручных клавишных и глокеншпиля [глокеншпиль – разновидность металлофона с набором колокольчиков или металлических пластин в качестве источника звука, – *прим. пер.*], типичных для начальных школ, в кабинете музыки было несколько старых инструментов, уцелевших с тех времен, когда формальное обучение музыке было более распространено. Трехструнные скрипки, потрепанные корнет-а-пистоны и заброшенное фортепиано времен короля Эдуарда [отца ныне здравствующий королевы] влились в микс. Фортепиано, по принципу позабытой Радиофонической мастерской BBC [The BBC Radiophonic Workshop, одно из подразделений для работы со звуковыми эффектами BBC, учрежденное в 1958 г. для создания эффектов и новой музыки для радио и закрытое в марте 1998 года, – *прим. пер.*], использовалось как клавишные и источник звуковых эффектов, на нем также учились играть мелодии и аккорды.

Интересно, что хотя у большинства членов клуба дома есть компьютер, а кое у кого даже свой личный, многие сроду не прикасались к пианино – может быть, современные малогабаритные жилища просто не вмещают столь громоздкие инструменты; но я подозреваю, что они просто сдались на милость повсеместно доступных готовых электронных развлечений. Дети на удивление охотно будут играть на скрипке, корнете или трубе, если просто дать им попробовать.

Одна из причин сравнительно беспрепятственной установки свободного ПО в том, что музыкальный клуб не входил в общий поток ИТ-снабжения – средства школы на него не тратились, и не было нужды пользоваться обычными каналами поставки, что могло привести совсем к другому результату. Школы не всегда хорошо принимают



Синтезатор *Alsa Modular Synth* на списанном компьютере стоит дешевле, весит меньше и выходит из строя гораздо реже, чем настоящий антикварный Moog Modular.



Вам не нужна драм-машина, если есть *Hydrogen* – да, он выглядит сложно, но дети его мигом осваивают.

Если бы братья Ферт из Эдмонтона знали, чем кончит их пианино, то что бы они подумали?

«советы посторонних» касательно организации ИТ, так как львиная доля получаемой ими информации исходит от консультантов и поставщиков, заинтересованных в развертывании новой Windows-системы, со всеми вытекающими последствиями.

Разные цели

Похоже, поставки нового ИТ-оборудования в школы слишком налегают на оборудование: крупные средства, потраченные на покупку новых компьютеров, внушают уважение к школе. С точки зрения непрофессионала кажется разумным, что раз компьютеры с Windows XP после трех-четырех лет работы ужасно тормозят, пора купить более новые и быстрые – а организации, поставляющие их в образовательные учреждения, и рады-радешеньки заставить это сделать.

И, как по волшебству, новые компьютеры известных марок с установленными Windows работают быстрее… в первые полгода или около того. Потом они неминуемо утонут в мусоре, который намеренно или нечаянно установят дети, когда им разрешат пользоваться этим компьютером для выхода в Интернет. Компьютер с Windows тихо умрет от удушья. Зато компьютеры с Linux почти не нуждаются в обслуживании и месяцев девять работают так же, как и в день установки; вот вам качество Debian.

Было бы куда лучше, если бы процесс обновления был «завязан» на приложения. Во-первых, завучи могли бы опросить учеников, какие программы им нужны или какие им хотелось бы иметь. Потом компьютеры, удовлетворяющие этим требованиям, можно собрать из имеющихся, еще до того, как школа потянется за чековой книжкой. Такой подход уделил бы больше внимания архитектуре системы и ее администрированию, чем цикл покупки–обновления парка. Сместились бы акценты в выделении ресурсов: вместо затрат на компьютеры и программы – найм сотрудников с должными знаниями. Наверное, не каждой начальной школе по карману опытный системный администратор, но несколько школ могут договориться, и один профессионал сможет контролировать четыре-пять установок.

Конечно, есть и действительно выгодные аппаратные обновления. ЖК-мониторы более эргономичны по сравнению с ЭЛТ, если только это не полная дешевка, и у них меньше энергопотребление. Но современные компьютеры используются в основном для web-приложений; было бы лучше применять новые процессоры для ресурсоемких задач, типа создания музыки и видео, а компьютеры постарше переделать в тонкие клиенты для работы в сети и доступа к серверным приложениям.

Это просто работает

Говоря о работе с творческими приложениями, жизненно важно обучить учителя. Ведь именно учителя – первый эшелон помощи детям, и нереально ожидать, что они будут на досуге сами вгрызаться в ПО. Дайана Гейтс [Diana Gates], учитель в школе Всех Святых, поддерживала идею клуба с самого начала, тратя на развитие творчества учеников свое личное время. Она комментирует: «Детям очень понравилось экспериментировать со звуком на компьютере. Это помогло им освоить ритм и побудило исследовать шаблоны и композицию».

Запись музыки на компьютере побудила учеников попробовать настоящие инструменты.

Этика свободного ПО хорошо сочетается с образованием. Например, при желании создать свой набор ударных инструментов для *Hydrogen* ученики начальной школы могут это сделать. В *Hydrogen* используется открытый формат, а файл настроек ударных инструментов – обыкновенный XML. Далее школа выложит этот набор на сайте и разделит его с другими, зная, что это не только законно, но и поощряется. Как любит подчеркнуть разработчик *Kamaelia* Майкл Спаркс [Michael Sparks], умение делиться – одна из первых вещей, которым учат детей, и передача в дар людям созданных нами музыки и файлов абсолютно естественна. LXF

Творческие дистрибутивы

Творческие приложения прекрасно устанавливаются в обычных дистрибутивах Linux, но есть несколько «нишевых» решений, предоставляющих готовую настольную систему для творчества. Некоторые из них особенно удобны для обработки звука в реальном времени, так как ограждают пользователя от необходимости латать ядро Linux или заменять его. Выбор здесь в основном зависит от того, какой из вариантов Linux вам наиболее симпатичен, так как большинство дистрибутивов – ремиксы.

64 Studio 2.1 основан на Debian Etch и ориентирован на производительность. Доступен в виде образа установочного DVD, есть также Live CD.
www.64studio.com

Dynebolic 2.5.2 – одна из пользовательских версий дистрибутива Linux From Scratch Live CD, она особенно удобна для старых ПК.
http://dynebolic.org

JackLab Audio Distribution 1.0 основан на OpenSUSE; доступен образ установочного DVD.
Live DVD в разработке
http://jacklab.net

Ubuntu Studio 8.04 стремится предоставить творческие приложения, максимально сохраняя внешний вид Hardy Heron.
http://ubuntustudio.org

Для компьютеров, имеющих привод CD-ROM, но не DVD-ROM, хороший выбор – Dynebolic. Другие дистрибутивы, наверное, можно установить из сети, если есть загрузочный CD. Большинство, если не все, дистрибутивы поддерживают запуск с USB-брелка, но современным ПК часто не хватает возможности загрузки с USB в BIOS, даже если USB-порты связаны с материнской платой.

В других дистрибутивах есть сторонние мультимедиа-репозитории, которые избавляют вас от необходимости собирать пакеты творческих программ из исходников. Пакеты музыкального ПО доступны на сайте Planet CCRMA At Home (http://ccrma.stanford.edu/planetccrma/software), а к Gentoo есть аудио-ориентированный оверлей.

Libre Graphics Meeting 2008

› В этом году конференция проводилась в прекрасном городе Вроцлаве.

LibreGraphics 2008

Талантливые художники, разработчики и **Ник Вейч** недавно объединили свои силы в честь третьей ежегодной встречи, посвященной графике – и вот что там творилось.

Солнечный Вроцлав стал в 2008 году местом проведения Libre Graphics Meeting (LGM), и в мае разработчики и сторонники графики с открытым кодом собрались на юге Польши. Выбор города оказался превосходным – улочки, благоухающие древней культурой и артистизмом (с изрядной дозой религиозного аромата) и обильно украшенные галереями, оперными театрами, просто театрами и скульптурами, хоть и рядом с жилыми зданиями в советском стиле.

Поиск места проведения конференции озадачил не одного гостя. Приглашающей стороной был Вроцлавский технологический университет, но этот адрес означает огромную территорию, практически целый прибрежный район города. Зато, добравшись туда, гости получали четыре дня всевозможных докладов – самая насыщенная и обширная из когда-либо существовавших программ LGM.

Здесь всегда бывает масса обсуждений, презентаций и прочих событий, посвященных четырем китам свободной графики – *Scribus*, *GIMP*, *Inkscape* и *Blender*. Причем Libre Graphics не только информирует, что поделывает ваше любимое графическое приложение, но и открывает нечто новое – поэтому особенно интересны появления людей из менее известных проектов. Как насчет *Hugin*? Это отличное приложение разрабатывается уже несколько лет; удивительно, что знают о нем очень немногие. Пабло д'Анджело [Pablo d'Angelo], ведущий автор, провел отличную презентацию, показав возможности программы по сшиванию панорам из фрагментов и генерации VR-изображений.

Пабло очень порадовало, что OpenGL-ускорение было выбрано одним из проектов Google Summer of Code: «Я думаю, это очень подкрепит нашу работу – и есть еще четыре принятых проекта». Дополнения, включающие улучшение распознавания облаков и пакетный рендеринг, должны сделать *Hugin* еще более впечатляющим, превратив его в достойную альтернативу коммерческим программам. Если вы с ним еще не знакомы, обязательно найдите его через свой менеджер пакетов, или загляните на сайт http://hugin.sourceforge.net.

› Уже догадались, что это? Пьер Маршан демонстрирует возможности *Fontmatrix*.

Photos: Andy Fitzsimon and Nick Veitch

› Присутствовало так много Eee PC и Blender, что это – самая типичная картина!

Одним из ярких пунктов в расписании Libre Graphics стал *Publican*. Кто не знает – сообщаем: это комплект от Red Hat на инструментарии DocBook/XML. Red Hat создала его главным образом для групповой разработки документов по собственным продуктам, но он, конечно же, вышел под лицензией GNU Public Licence для всеобщего пользования. С помощью нескольких обманчиво простых команд вы можете создать полную рабочую структуру DocBook и вывести результат в формат PDF или HTML – что вам больше нравится.

Publican

Энди Фитцсаймон [Andy Fitzsimon] из Red Hat провел ознакомительную презентацию *Publican* (в свет его вывела Fedora 9 – и это, наверное, наилучшая для него платформа) и превознес его достоинства как средства коллективного документирования. Имеются RPM, и, судя по всему, поддержка других пакетов и дистрибутивов не заставит себя ждать. *Publican* вряд ли станет кросс-платформенным, поскольку он опирается по большей части на существующие инструменты Unix, что весьма необычно для программы, представленной на LGM. Энди Фитцсаймон постарался, чтобы разработчики в зрительской аудитории тепло приняли *Publican*: «Его миссия – отдать должное свободному ПО и объяснить про него людям. Я лично знаю таких, кто и не взглянет на программу, если она не снабжена правильной документацией, а имея такой прекрасный инструмент [разработчика], обидно упускать пользователей».

По части управления шрифтами, история Unix несколько напоминает шахматную доску, и это обусловило множество дискуссий. Хотя совместимость с форматами шрифтов существует уже давно, реально серверы шрифтов несколько архаичны и сложны. Установка шрифтов стала проще, чем раньше, но управление ими по-прежнему остается проблемой, особенно для профессионалов дизайна на системах Linux. Но выход есть! Один из невоспетых героев среди инструментов из арсенала дизайнера – *Fontmatrix*.

Эта программа – аналог приложений типа *Adobe Type Manager* или *Extensis Suitcase*. Она просканирует систему на предмет наличия шрифтов, покажет найденные и позволит пользователю включать и отключать их. Шрифты можно маркировать или группировать, чтобы привязать к документу или проекту. А зачем нужно отключать шрифты? Если их у вас тысяча, то во многих приложениях меню шрифтов становится слишком громоздким, а у *GIMP* на загрузку уходит полчаса! Но в *Fontmatrix* есть не только включение и отключение шрифтов – он также хорош для их предварительного просмотра, а в ближайшее время функций в нем еще прибавится. Один из главных энтузиастов *Fontmatrix*, Питер Линнел [Peter Linnell] из *Scribus*, говорит: «Я занимался допечатной подготовкой, и менеджер шрифтов – это очень важный инструмент. При неправильной работе он может бесконечно осложнить вашу жизнь. Удивительно, сколько он [Пьер Маршан – Pierre Marchand, разработчик *Fontmatrix*] в него вложил. Он проникает в самое нутро шрифтов – такого я не видел еще ни в одной программе, кроме некоторых редакторов шрифтов. Ничуть не удивлюсь, если его-то и добавят следующим». »

> «Fontmatrix — один из невоспетых героев в арсенале дизайнера.»

› Виртуальные Evilnick (слева) и Romanjoost (справа) наконец-то встретились – их разделяет только Жоао Бузно [Joao SO Bueno].

» Ойвинд Колас [Oyvind Kolas], более известный сообществу *GIMP* как Пиппин [Pippin], явился, чтобы продемонстрировать последний релиз *Gegl* – Generic Graphics Library. *Gegl* – это графический движок, задуманный как будущая основа для *GIMP*, но у него есть и своя жизнь.

> «Подобно всем хорошим разработчикам, Карл Ворт сам утолил свою жажду.»

Имея привязки для множества языков, хорошую поддержку глубины цвета и несколько интригующих функций, плюс простоту расширения, он обречен на популярность и вне *GIMP*. На презентацию Коласа собралась масса сочувствующих, поскольку верные поклонники *GIMP* оказались на LGM солидной силой.

Krita и Cairo

Krita также была представлена на Libre Graphics Meeting. Благодаря поддержке палитры CMYK, она долгое время была спутницей типографских дизайнерских работ, которую предпочитали *GIMP*, пока что лишенному данной функции. Но *Krita* – не просто бедная родственница с единственным достоинством, что показала сделанная презентация по иллюстрации natural media [средства изображения естественной среды, – *прим. пер.*].

Графическая библиотека Cairo родилась как результат разочарования Карла Ворта [Carl Worth] в имеющихся альтернативах. Подобно всем хорошим разработчикам, он утолил свою жажду, создав библиотеку, ныне ставшую кросс-платформенной и используемой во множестве проектов – хотя и не всегда с наилучшим эффектом. Ворт вызвал у аудитории самый громкий смех, показав реализацию Cairo в *OpenOffice.org*. Слайд «до» показывал круг в увеличении: кривой, с выбившейся за края заливкой и проплешинами внутри. Слайд «после» демонстрировал все ту же корявость, но на сей раз… после сглаживания! Кое-кто среди зрителей покраснел, так что, полагаем, в будущем *OpenOffice.org* будет лучше использовать Cairo.

Амбиции Ворта относительно проекта до сих пор остаются неудовлетворенными, даже при почти что глобальном доминировании: «Я хочу, чтобы его использовали или, по крайней мере, могли использовать во всех графических проектах. Многое еще предстоит сделать. Хочу добавить поддержку PDF, это было бы неплохо для применения в [таких проектах, как] *Scribus*», сообщил он нам. Это будет непростая задача, ведь многие считают, что у *Scribus* лучшая реализация PDF после самой Adobe (а некоторые даже уверяют, что *Scribus* ближе к стандартам!), но слушатели Ворта идею охотно поддержали.

Послевкусье

Выпивка с пиццей в компании избранных представителей эффектно завершила деловую часть Libre Graphics Meeting для вашего корреспондента *Linux Format*, но даже во время заключительной дринк-сессии вечером было никуда не деться от разговоров о рендеринге шрифтов и реализациях UTF. И хотя место проведения нелегко было отыскать, Libre Graphics Meeting 2008 во Вроцлаве определенно стал самым интересным из всех. Куда в следующий раз? Вроде бы что-то там говорилось про Сингапур. LXF

Почетные гости

Было бы неправильно не упомянуть Дэйва Нири [Dave Neary] и Йона Филипса [Jon Philips]. Правда, они не выступали на самом мероприятии, но именно их работа по изысканию финансового обеспечения конференции позволила провести LGM. Спасибо, парни, надеюсь, увидимся в следующий раз!

» Энди Фитцсаймон недоумевает: Eee PC – это самый крошечный портативный компьютер или самый большой мобильный телефон?

Libre Graphics Meeting 2008

Scribus

Участники команды *Scribus*, похоже, трудолюбивее всех. Даже если вы не заметили особых исправлений или добавок по сравнению с прошлым годом, работа над проектом проделана очень большая. И даже на LGM они не теряли ни минуты! Если команда *Scribus* не участвовала в разговорах и не давала презентаций, они тут же собирались вместе и принимались обсуждать и работать над тем, что должно быть сделано в следующую очередь.

Самая свежая версия *Scribus* использует библиотеку Cairo. «Она позволяет делать очень интересные вещи с линиями и векторными форматами, и потрясающе увеличивает скорость», говорит Питер Линнел. Он также рассказал о новой библиотеке для работы с PDF – она позволит *Scribus* поддерживать более широкий спектр графики.

GIMP

Большинство дискуссий по поводу *GIMP* касалось не серии 2.4, а грядущего релиза 2.5.1. В некоторых операциях он опирается на *Gegl*, и есть идея использовать в будущем *Gegl* гораздо шире. *Gegl* значительно увеличит скорость, а также обеспечит поддержку некоторых вещей, которых *GIMP* очень не хватало – большую разрядность цвета и количество палитр.

Однако не все изменения касаются графического движка и скрыты внутри: многие инструменты исправлены в плане эргономичности, и у них у всех усовершенствованы производительность и точность.

Inkscape

В этом году разработчиков *Inkscape* на Libre Graphics Meeting было маловато, поскольку многие не смогли приехать. Но их интересы рьяно отстаивал Энди Фитцсаймон (да-да, тот, что представлял *Publican*!), и мы увидели отличную демонстрацию недавно выпущенной версии 0.46. Обладая поддержкой разных цветовых палитр, PDF-графики, улучшенной работой с SVG и всеми новыми эффектами, *Inkscape* очень быстро наращивает свои силы.

Один из умопомрачительных эффектов, находящихся на данный момент в разработке – это поддержка 3D-объектов, которые можно импортировать и векторизовать почти так же, как cell-shader в программе трассировки лучей. Это определенно улучшит качество рисунков многих пользователей!

Blender

Посетителей LGM угостили *Big Buck Bunny*, самым свежим мультфильмом от Blender Foundation. Этот показ был значимым, поскольку продемонстрировал важные усовершенствования в *Blender*, разрабатываемые на стыке с анимацией. Самих разработчиков на LGM тоже было не в избытке: они спешат завершить релиз 2.46, который включит значительно улучшенную имитацию меха и ворсистой ткани и слияние с окружающей средой – это некая разновидность эффекта мягкого самозатенения, быстро вводящего текстуру и детали. Система частиц полностью пересмотрена; в основном поправки коснулись меха и волос, но и движок обзавелся более правильной физикой.

Что за штука...

TimeVault?

Давайте сверим часы и присоединимся к **Майку Сондерсу** в путешествии во времени по файловой системе Linux...

TimeVault – напоминает о капсулах с посланием для потомков... Я прав?

[Vault – склеп, а также сейф, хранилище, – *прим. пер.*] Да-да, это как раз Linux-тема. Капсула захоронена под крупнейшей колонией пингвинов на Земле королевы Мод в Антарктике, в неё заложена первая версия ядра Linux, комплект дисков Slackware 2.0 и клок бороды Ричарда Столлмана.

А если без шуток – что это такое?

TimeVault – система резервного копирования для Linux, подобная *Time Machine* из Mac OS X 10.5. Обычно такие программы попросту берут кусок файловой системы и кидают его на внешний носитель. *TimeVault* гораздо умнее: она отслеживает изменения в файловой системе, не дублируя файлы. Ведь, как правило, большинство наших данных меняется не каждый день (и даже не каждую неделю). Мы работаем, от случая к случаю копируем новую музыку, но основная часть нашего добра лежит нетронутая.

> «У TimeVault, как и у Time Machine, припасен особый фокус.»

А откуда программа узнаёт, какие файлы меняются, удаляются и добавляются?

При установке *TimeVault* создаёт фоновый процесс (демон), а тот следит за состоянием директорий, указанных пользователем. Если один из файлов в заданном каталоге меняется, программа делает его копию и сохраняет в условленном месте, чтобы при необходимости старую версию можно было восстановить. Например, вы испортили фото в *GIMP* или поработали с кодом, и он перестал компилироваться; *TimeVault* сможет воссоздать файл таким, каким он был вчера, неделю и даже месяц назад, причём время и ресурсы CPU зря не тратит: программа настраивается на любую периодичность, от нескольких минут до недели. Директории на бойком месте (скажем, той, где коды ваших программ) подойдёт ежечасное сохранение. А вот семейные фото вы вряд ли часто переделываете, этой папке довольно и раза в неделю.

И много памяти требует резервная копия?

Это зависит от типа файлов. Если вы редактируете видео, места понадобится много – сотня, а то и больше гигабайт. Тексты или программы будут поскромнее. Но регулярное копирование сейчас выгодно как никогда, ведь цены на накопители стремительно падают. Жесткий диск на 160 ГБ стоит 1000 – 1500 рублей; если этого мало, купите 1 ТБ за 5000. Этого вам надолго хватит.

Ну, хорошо. Допустим, мне нужно регулярное резервирование. У меня 10 ГБ MP3, и я поручаю *TimeVault* сохранять их еженедельно. Значит, каждую неделю ему подавай 10 ГБ?

Да что вы, нет! *TimeVault* не транжира: как и в *Time Machine*, тут есть особый фокус. Вместо того чтобы просто копировать содержимое директории на внешний носитель, программа сравнивает версии файлов, уже имеющиеся в архиве, с текущими. Пусть ваши 10 ГБ музыки копировались на прошлой неделе; в следующий раз *TimeVault* проверит содержимое директории и запишет только новые и изменённые файлы.

Это сберегает массу дискового пространства. Первая резервная копия, конечно, будет увесистой, но последующие потребуют только записи изменений, а не копирования всех файлов. Конкретные числа могут различаться, но в среднем для ежедневного резервирования работы последних трёх месяцев должно хватить диска умеренной ёмкости (гигабайт 50).

Ну, а как вернуться к старым копиям файлов?

TimeVault состоит из двух частей: фоновый процесс-демон следит за состоянием директорий и в указанное время делает их снимки. Другая часть – графический интерфейс, в нём происходит настройка программы и выбор каталогов для сохранения. Пока что разработка *TimeVault* сосредоточена в сообществе Ubuntu, поэтому GUI имеется только для Gnome. Обсуждение KDE-варианта уже идёт, и есть надежда, что ради прогресса Linux в целом программа со временем станет универсальной.

В системном лотке на рабочем столе Gnome поселяется значок, который уведомляет вас об изменении файлов. Через него можно запустить Snapshot Browser, программу вроде файлового менеджера, которая позволяет просматривать содержание предыдущих копий. Грохнули вы, к примеру, любимую MP3-запись или сделали роковую ошибку в каком-нибудь документе – Snapshot Browser повернёт время вспять, выбрав определённый день и час, и вы получите прежний файл, как ни в чём не бывало. Просто сногсшибательно – даже если вы сидите!

Чудесно. Значит, я указываю *TimeVault* на свою домашнюю директорию, и всё будет в шоколаде?

Э, нет! Домашняя директория – статья особая. Здесь активность файловой системы чрезмерна. Попробуйте набрать **ls -a** в терминале – вы увидите множество скрытых файлов и директорий, названия которых начинаются с точки. Это временные файлы или настройки программ, и они меняются чуть ли не ежеминутно, а то и чаще. Возьмём *Firefox*. Файлы его настройки хранятся в **.mozilla/** в вашей домашней директории. Кэш-файлы хранятся тут же – а ведь при web-сёрфинге они меняются непрерывно. Если поручить *TimeVault* сохранять домашнюю директорию, программа будет работать только на кэш *Firefox*, и дисковое пространство улетучится со свистом.

Лучше немного подумать и задать *TimeVault* сохранение только тех директорий, в которых хранятся музыка, фото, коды программ и т.п. Регулярные снимки всей домашней директории разорительны.

А какие новые функции планируются в будущих версиях *TimeVault*?

Как насчёт API *TimeVault*? С его помощью сторонние программы смогут задействовать функциональность *TimeVault* и избежать лишнего кодирования. Один из примеров, приведённых разработчиками – почтовый клиент *Evolution*. Представьте, как удобно прокрутить список предыдущих сообщений и отыскать среди них

случайно удалённые. То же относится к списку контактов, событиям в календаре и другим данным, хранимым программой.

Рассматриваются два подхода к такому взаимодействию. Первый – заставить *TimeVault* работать как файловый сервер, чтобы по запросу другие программы сохраняли и восстанавливали свои данные на определённые даты. Второй – программы могли бы запрашивать запуск GUI для просмотра резервных копий; хотя тут трудно добиться такой гибкости, чтобы способ был применим ко всем типам приложений.

Текущий план – добавить два новых вызова к настольной системе коммуникации D-BUS. «Озабоченная» программа сможет обратиться к *TimeVault* с предложением: «сохраняй данные такого-то участка файловой системы с такой-то частотой (ежечасно, ежедневно и т.д.)». И наоборот, при необходимости приложения смогут пользоваться хранилищами *TimeVault* с целью восстановления утраченных данных. Мы, как конечные пользователи, тоже сможем просматривать эти хранилища, ограничивая вывод с помощью системы фильтров.

Как всё это будет работать на практике, мы ещё поглядим, но связка *TimeVault* с такими китами, как *Evolution*, *OpenOffice.org* и *Scribus*, даст настольному Linux немалые козыри.

Ладно, хватит про функции. Дайте, дайте скорее адрес!

Ах я тормоз! Как уже упоминалось, разработка *TimeVault* связана с Ubuntu и Gnome, поэтому для тестирования программы необходимо установить именно этот дистрибутив. А вот фоновому демону он безразличен, и если вам пришла охота заняться хакерством, ступайте прямо на **http://wiki.ubuntu.com/timevault** – там ждут помощи в портировании программы.

Если у вас установлен Ubuntu (пусть в виртуальной машине), можете взять Deb-пакет последней версии с вышеуказанного адреса и попробовать его. Проект пока в альфа-стадии, так что серьёзные данные доверять ему не стоит, но дело идёт к версии 1.0, и уже сейчас результаты впечатляют. Надеюсь, мы увидим *TimeVault* уже в следующей версии Ubuntu, а если наладится портирование – и в очередном цикле прочих дистрибутивов. LXF

По рецептам д-ра Брауна

Д-р Крис Браун
Доктор обучает, пишет и консультирует по Linux. Ученая степень по физике элементарных частиц ему в этом совсем не помогает.

Добротное администрирование систем из причудливых заворотов кишок серверной.

Дисбаланс учебных курсов

Вчера у меня на столе оказался список курсов ведущей организации по обучению в сфере ИТ. В нем – 22 учебных курса со словом 'Windows' в названии и четыре – с 'Linux'. Пошарив по сайтам аналогичных компаний, я увидел то же самое. В списке второй компании было 50 названий с Windows и 11 с Linux. У третьей – девять курсов по Windows и два по Linux. Пропорция в каждом случае примерно пять к одному. Если посмотреть на соотношение приложений Windows/Linux, дисбаланс усиливается. Одна из компаний предлагает 15 учебных курсов по *SQL Server* и ровно ноль по *MySQL*. И наверняка учебных курсов по *Internet Information Services* больше, чем по *Apache*, несмотря на то, что доля *Apache* на рынке более чем на 50% превышает долю *IIS*.

Я не сравниваю, к примеру, частоту, с которой проводятся занятия, и не оцениваю среднее количество бездельников, их посещающих. Я просто сопоставляю количество учебных курсов, которые должны покрыть две технологические области.

Но почему?

Почему так? Потому что Linux проще, или там нечего изучать? Сомневаюсь. Потому что ПО свободное, и люди считают, что суммарная стоимость владения Linux должна быть равна нулю, а следовательно, не хотят платить за учебные курсы? Возможно. Но я подозреваю, что люди, которые пользуются Linux и, например, создают сайты на базе LAMP (Linux, *Apache*, *MySQL* и PHP), обладают особенно активным складом ума, и лучше разберутся во всем сами, чем будут сидеть в классной комнате и слушать объяснения педагога. Я и правда не знаю. Кабы знал, пожалуй, разбогател бы.

Блоки в коробке

Распределенное реплицируемое блочное устройство

Низкоуровневая репликация файловой системы Linux

Несколько месяцев назад в **LXF101** и **LXF102** я писал о построении кластеров – как высоконадежных (с использованием *Heartbeat* для обработки отказов), так и высокопроизводительных с балансировкой нагрузки через *LVS* (*Linux Virtual Server*). Но я не затронул одну из проблем: создание отказоустойчивой файловой системы. Недавно мое внимание привлекла технология DRBD (Distributed Replicated Block Device – распределённое реплицируемое блочное устройство). Блочные устройства в Linux – это хранилища, данные с которых могут считываться и записываться блоками в произвольном (в отличие от последовательного) порядке; такие устройства важны, поскольку Linux использует их для построения файловых систем. В Linux есть несколько технологий, использующих механизм отображения для создания дополнительного слоя блочных устройств поверх физического накопителя. Одна из них – логические диски, другая – программный RAID, третья – зашифрованные файловые системы.

DRBD – распределенная система хранения. Она определяет основной и резервный серверы и зеркалирует содержимое блочных устройств основного сервера

› Локальный жесткий диск – классический пример блочного устройства.

на резервный. Концептуально это похоже на RAID 1, с той разницей, что выполняется по сети. При нормальной работе системы операции чтения и записи просто передаются на соответствующее физическое устройство (диск) основного сервера. Операции записи также отправляются по сети на резервный компьютер, где зеркалируются. Если основной компьютер выходит из строя, резервный замещает его. При возврате основного компьютера в сеть DRBD применяет эффективный механизм синхронизации, обновляя только те блоки устройства хранения основного сервера, что претерпели изменения за время его простоя.

DRBD написали Ларс Элленберг [Lars Ellenberg] и Филипп Райзнер [Philipp Reisner]. Это продукт производителя открытого ПО Linbit (**www.linbit.com**). Он не особенно новый, но лишь недавно включен в популярные версии ядра Linux, и будет любопытно поговорить о нем поподробнее в следующих номерах Linux Format. Следите за новостями!

Совет дня

Примерно раз в неделю я беру свой коврик для мыши (такой с валиком под запястье), мою его в мыльной воде и слегка посыпаю тальком. Попробуйте. Мышь полюбит вас навсегда.

Сначала была предустановка

Kickstart Хорошая новость для измотанных системных администраторов: утилита автоматической инсталляции Red Hat теперь есть в Ubuntu.

Если вы устанавливали Linux лишь однажды, этот опыт наверняка показался вам любопытным. Даже во второй или третий раз нажимать на кнопки интересно. Но если вы работаете в ИТ-отделе, занятом вводом в строй сотен компьютеров, ручная установка уже отнюдь не радует. В идеале хотелось бы достать компьютер из коробки, подсоединить его к сети, включить и выйти вон, зная, что через час Кларисса, новая сотрудница Отдела контроля гигиены, сможет зайти в офис и запустить свой свеже-дезинфицированный компьютер.

Ubuntu очень много унаследовал от Debian, и для автоматизации установки в прошлом использовал технологию «предустановки» [preseeding]. Здесь все без сюрпризов. Поэтому я слегка удивился, обнаружив, что в Hardy Heron предпочли другой способ автоматизации установки – *Kickstart* (изначально, приложение Red Hat). Обе технологии служат одной основной цели – обеспечить «пассивную» инсталляцию путем указания (в файле) ответов на вопросы, обычно задаваемые установщиком во время загрузки. (Существуют и другие подобные технологии. В SUSE используется *AutoYast*, и задолго до всех у Sun Microsystems был *JumpStart*.) Самый простой способ создать файл *Kickstart* для Ubuntu – воспользоваться графической утилитой *system-config-kickstart*.

Результат работы **system-config-kickstart** – файл конфигурации *Kickstart*, по соглашению названный **ks.cfg**. На экранном снимке (внизу) показан его фрагмент. К сожалению, окно выбора пакетов в *Kickstart* сейчас, кажется, не работает, и их названия придется отредактировать в файле вручную. Однако во многих случаях хватит и двух следующих строк:

```
%packages
@ ubuntu-desktop
```

Одна из действительно интересных штук в версии *Kickstart* для Ubuntu в том, что можно передавать параметры предустановки инсталлятору Debian, используя ключевое слово ‘preseed’ в файле конфигурации *Kickstart*. Это дает системному администратору большие возможности для настройки установки, чем *Kickstart* сам по себе. Например, данная строка в **ks.cfg** добавит предустановку для определения пароля root в *MySQL*:

```
preseed mysql-server/root-password password victoriassecret
```

Установка с Kickstart

Чтобы провести установку с *Kickstart*, файл **ks.cfg** должен быть «виден» во время инсталляции на целевом компьютере. Для этого положите его на сетевой сервер, доступный по FTP, HTTP или NFS. Альтернатива – модифицировать ISO-образ установочного диска, включив туда файл *Kickstart*. А можно даже записать файл на дискету.

Загрузив целевую систему, запустите установку *Kickstart*, указав параметр **ks=**. Вот, например, как инсталлируется система с файла *Kickstart* на CD:

```
ks=cdrom:/path/to/ks.cfg
```

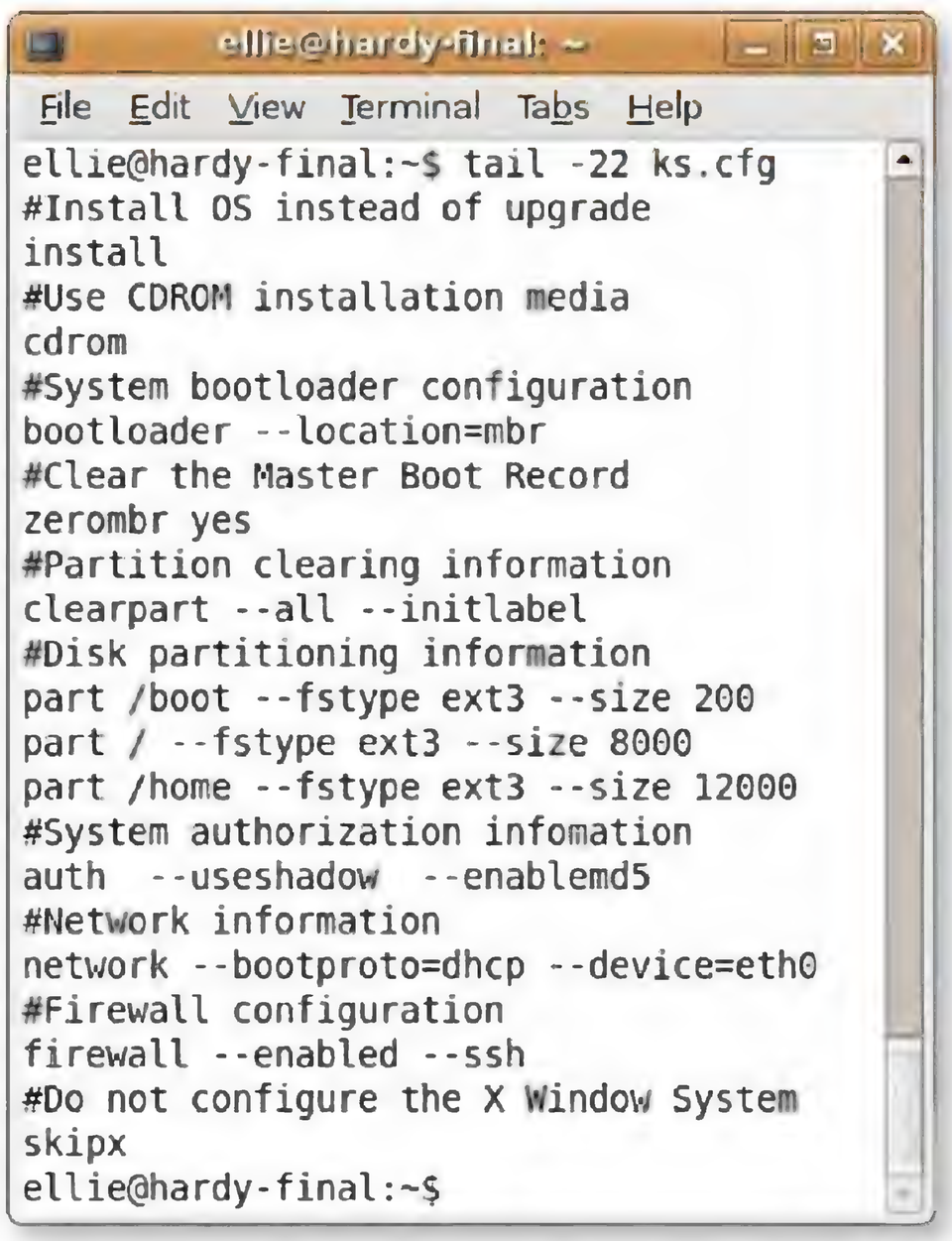
Если файл *Kickstart* на дискете, строка примет вид

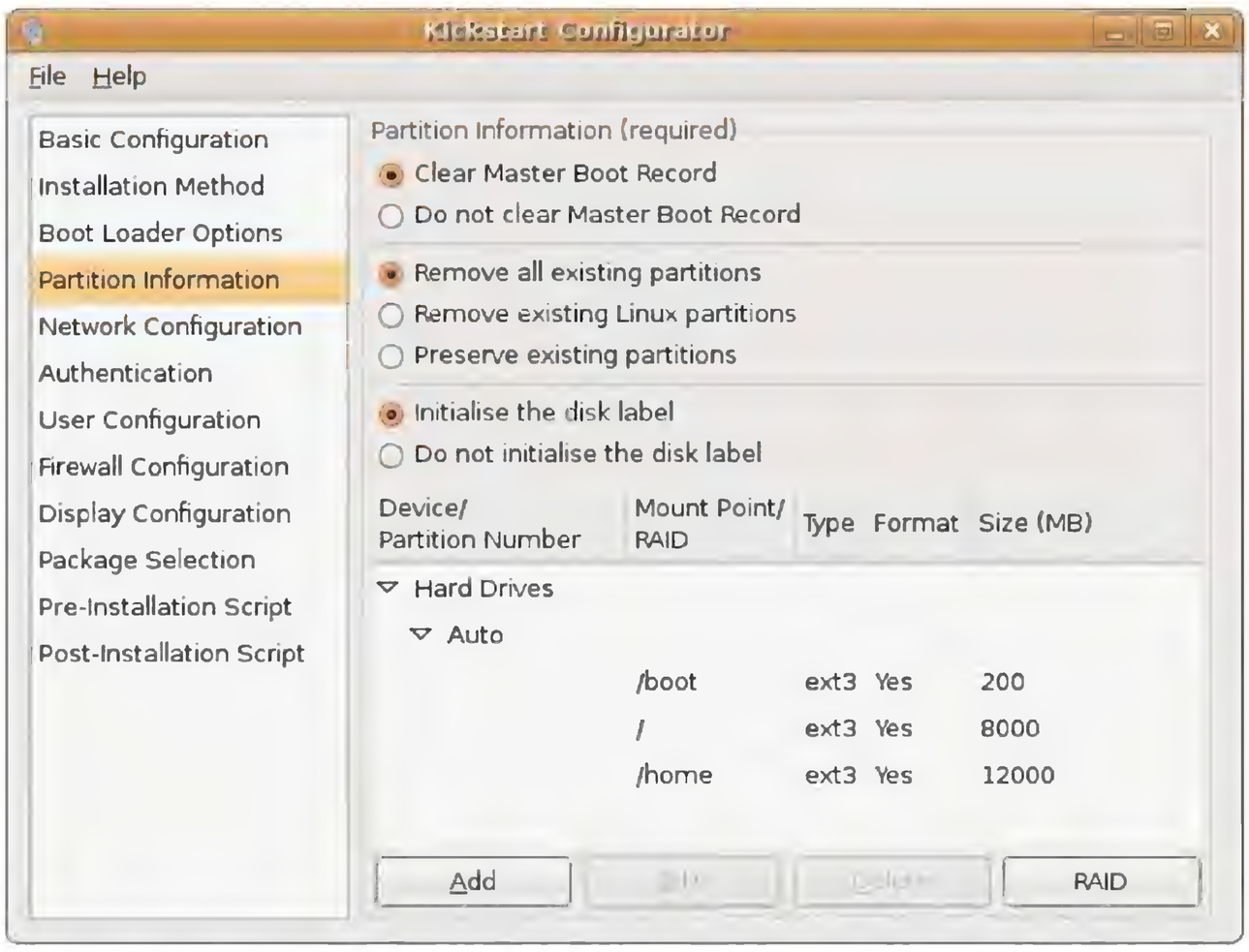
```
ks=floppy:/path/to/ks.cfg
```

а если он на web-сервере –

```
ks=http://url/to/ks.cfg
```

Теперь вы легко отделаетесь от Клариссы, запустив ее компьютер с нового образа диска; но чаще автоматизированная установка используется, наверное, в коммерческом дата-центре для добавления новых серверов, особенно если их нужно установить целую стойку. Чтобы это сделать, целевой компьютер нужно загружать в режиме PXE. Он получит IP-адрес и информацию о загрузке от DHCP-сервера, его ядро – с TFTP-сервера, конфигурацию *Kickstart* – с HTTP-сервера, а приложения – из репозитория пакетов. »





› (Внизу) **Один из 11 экранов, позволяющих администраторам редактировать файл конфигурации *Kickstart*. Здесь мы определяем разбиение диска.** (Внизу справа) **Часть файла конфигурации *Kickstart* с теми же данными о разбиении диска.**

Юный выскочка

init малость устарел; пришло время нового демона запуска системы.

Много лет в Linux использовалась схема управления запуском сервисов во время загрузки, которая появилась в System V Unix еще в 1983 году. За эти годы системные администраторы подружились со структурой System V и уровнями выполнения. Они знают и о скриптах запуска сервисов в **/etc/init.d**, и обо всех этих символических ссылках с названиями на 'S' и 'K', и о процессе *init* с его файлом конфигурации **/etc/inittab**.

Поэтому исчезновение **inittab** в некоторых последних дистрибутивах Linux немного шокирует, как и замена почтенного *init* на *upstart*. Первым дистрибутивом, выполнившим ее, был Ubuntu (еще в Edgy Eft более полутора лет назад). В Fedora 9 также используется *upstart* (это позволяет предположить, что за ним со временем последует Red Hat Enterprise Linux), а разработчики OpenSUSE 11.0 предпочли остаться со старым добрым *init*.

Upstart – это основанная на событиях замена *init*; управляет запуском задач и сервисов во время загрузки и их остановкой во время выключения системы и наблюдает за ними во время работы. Демон *upstart*, для обратной совместимости (и для вящей путаницы) названный *init*, управляется «событиями», формируемыми D-BUS. У события есть простое текстовое имя, например, 'startup', 'control-alt-delete' или 'runlevel'.

При запуске демон читает все файлы в **/etc/event.d/**. Каждый из них определяет задание, которым будет управлять *upstart*. Имя задания совпадает с именем файла. Эти файлы устанавливают, какие события вызовут запуск (или останов) задания и что делать, когда происходит событие. Вот простой пример – файл события Ctrl+Alt+Delete из Fedora 9 (обе его строки!):

```
start on control-alt-delete
exec /sbin/shutdown -r now "Control-Alt-Delete pressed"
```

Когда задание активизируется, оно проходит через серию состояний: ожидание, запуск, выполнение, остановка и т.д. Эти изменения состояний сами порождают события: «запускается», «запущено», «останавливается» и «остановлен». Поэтому можно определить задание, запускаемое по окончании другого. Вот файл события **prefdm** – он отвечает за запуск графического экрана входа в систему сразу после перехода на уровень выполнения 5 (т.е. после завершения задания 'rc5'):

```
start on stopped rc5
stop on runlevel [!5]
console output
respawn
respawn limit 10 120
exec /etc/X11/prefdm -nodaemon
```

Директивы **respawn** в этом файле говорят, что *upstart* должен перезапустить программу после ее завершения, но не более 10 раз каждые 120 секунд. Механизм обработки событий *upstart* предоставляет большую гибкость в определении того, что должно произойти при выполнении заданных условий; можно, например, потребовать «смонтировать USB-брелок, когда он обнаружен». Впрочем, ни в Ubuntu, ни в Fedora я не увидел подтверждения тому, что *upstart* настроен на такие штуки.

Upstart поставляется с утилитой *initctl,* позволяющей получить список заданий, показать все сгенерированные события и переходы и даже вызвать собственные события вручную. Есть и утилита *telinit*, предоставленная для обратной совместимости, хотя, насколько я могу судить, она просто вызывает событие подходящего уровня выполнения.

Поэкспериментируем

Понять, как работает *upstart*, поможет простой эксперимент. (Я проводил его на Fedora 9, но все должно работать точно так же на Ubuntu.) Начнем с создания файла **sayhello** в каталоге **/etc/event.d** с таким содержимым:

```
start on greet
console output
script
        echo "hello $1" at $(date) > /tmp/upstart-test
end script
```

Затем просигнализируем *upstart* перечитать файлы событий. Как и у старины *init*, у демона *upstart* идентификатор процесса равен 1:

```
# kill -HUP 1
```

Теперь проверим, что *upstart* знает о нашем задании, с помощью команды:

```
# initctl list
control-alt-delete (stop) waiting
logd (stop) waiting
prefdm (start) running, process 2119
rc0 (stop) waiting
... кое-что опущено ...
sayhello (stop) waiting
... опущено еще больше ...
tty5 (start) running, process 2123
tty6 (start) running, process 2127
```

Вызовем событие 'greet' с помощью *initctl* таким образом:

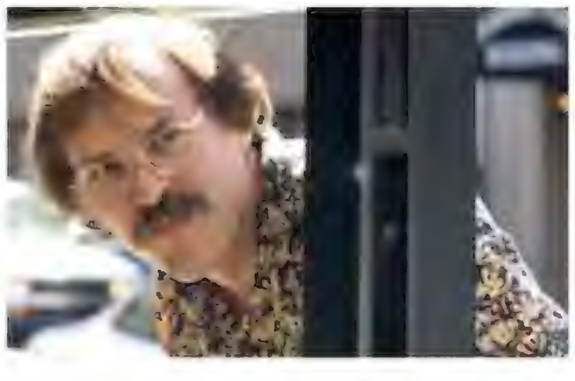
```
# initctl emit greet ellie
greet ellie
sayhello (start) waiting
sayhello (start) starting
sayhello (start) pre-start
sayhello (start) spawned, process 9333
sayhello (start) post-start, (main) process 9333
sayhello (start) running, process 9333
sayhello (stop) running
sayhello (stop) stopping
sayhello (stop) killed
```

Не цитируйте меня...

Мой коллега, преподаватель по Linux, как-то спросил меня: «Чем различаются теория и практика?» Как, наверное, требует профессиональная этика в такие моменты, я ответил: «Не знаю, а чем?» «Ну, – сказал он, – в теории разницы нет, а на практике есть».

Компьютерщики привыкли присваивать слова: ремешок ботинка (начальная загрузка – bootstrap), семафор, сигнал, молотить (thrash), мышь, печенье (cookie – куки), пламя (flame – флейм), мясные консервы (spam – спам), вилка (fork) и т.д. – и это напоминает мне наблюдение Шалтая-Болтая из *«Алисы в Зазеркалье»* Льюиса Кэрролла. Он сказал: «Когда я употребляю слово, оно означает только то, что я хочу, чтобы оно означало – ни больше, ни меньше». Вот моя любимая цитата из Ларри Уолла [Larry Wall]: «Скажем, вы вернулись в прошлое к Аде Лавлейс [Ada Lovelace, дочь поэта Байрона; считается самым первым программистом, – *прим. ред.*] и спросили бы ее, в чем разница между скриптом [сценарием] и программой. Она бы, наверное, улыбнулась и сказала что-то типа: "Ну, сценарий дают актерам, а программку – публике"».

И моя самая любимая, из Роберта Ферта [Robert Firth]: «Одной из главных причин падения Римской Империи было отсутствие нуля в нумерации: не было способа узнать, что их программы на C успешно завершились».



```
sayhello (stop) post-stop
sayhello (stop) waiting
```

Вывод команды ясно показывает изменения состояний, порождаемые нашим заданием ‘sayhello’. А проверка содержимого файла **/tmp/upstart-test** должна дать соответственно:

```
# cat /tmp/upstart-test
hello ellie at Mon Jun 2 06:45:36 BST 2008
```

Чтобы продвинуться на этап вперед, определим второе задание ‘saygoodbye’ в файле **/etc/event.d**. Оно будет запускаться после завершения ‘sayhello’:

```
start on stopped sayhello
console output
script
        echo “goodbye world” >> /tmp/upstart-test
end script
```

Конечно, будет нужно велеть *upstart* перечитать файлы. Теперь откройте второе окно терминала. Выполните в нем следующую команду:

```
# initctl events
```

Она останется работать в фоне и сообщит обо всех событиях.

Вернитесь в первое окно и сгенерируйте другое событие ‘greet’:

```
# initctl emit greet taylor
```

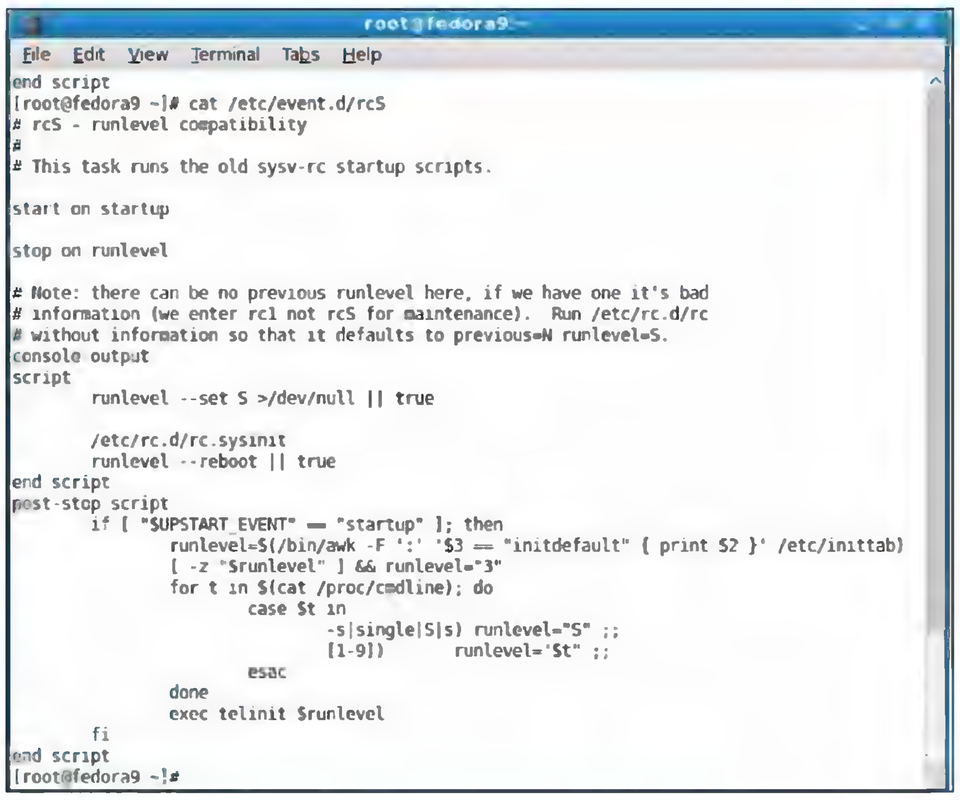
потом вернитесь во второе, чтобы посмотреть, какие события были сгенерированы:

```
greet taylor
starting sayhello
started sayhello
stopping sayhello ok
stopped sayhello ok
starting saygoodbye
started saygoodbye
stopping saygoodbye ok
stopped saygoodbye ok
```

и в завершение эксперимента снова проверьте содержимое файла **/tmp/upstart-test**, чтобы убедиться в том, что задание ‘saygoodbye’ запускалось:

```
# cat /tmp/upstart-test
hello taylor at Mon Jun 2 07:01:45 BST 2008
goodbye world
```

Ну да, пример игрушечный. Но я надеюсь, что он помогает почувствовать, как работает механизм обработки событий *upstart*, и показывает некоторые его возможности. Он мог бы заменить, например, запуск задач по расписанию, который сейчас выполняется *Cron*. Однако его главная цель на данный момент – просто заместить *init* как средство запуска сервисов во время загрузки. В самом *upstart* нет уровней выполнения, но просмотрев файлы событий в **/etc/event.d**, можно увидеть, что конфигурация эмулирует прежнее поведение. Все начинается с события startup, активизирующего задание rcS, которое выполняет традиционный низкоуровневый скрипт инициализации **/etc/init.d/rcS** (в Ubuntu) или **/etc/rc.d/rc.sysinit** (в Fedora). Что произойдет дальше, зависит от конкретного дистрибутива.



› Файл событий /etc/event.d/rcS в Fedora 9 содержит довольно искусный код для обратной совместимости с традиционной моделью уровней выполнения.

Потенциал

В Fedora 9 задание rcS запускает скрипт ‘post-stop’ – он вытаскивает уровень выполнения по умолчанию из **/etc/inittab** и вызывает *telinit* с этим уровнем выполнения как аргументом. Версия *telinit* в *upstart* нужна только для обратной совместимости, она просто генерирует событие ‘runlevel 3’, которое в свою очередь активизирует задание rc3, запускающее традиционный скрипт **/etc/rc.d/rc3** для перехода на желаемый уровень выполнения.

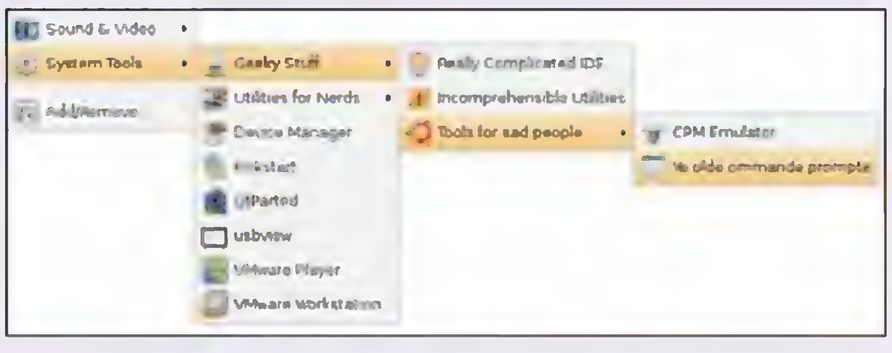
Чтобы разобраться с Ubuntu, потребовалось чуть больше времени, но я в конце концов заметил файл **rc-default**, запускаемый по завершению задания rcS (он содержит строку ‘start on stopped rcS’). Этот файл также пытается получить уровень выполнения по умолчанию из **/etc/inittab**, а если такого файла нет (кстати, в установке по умолчанию его и нет), то переходит на уровень выполнения 2 путем вызова *telinit*. Но и в Ubuntu, и в Fedora все заканчивается вызовом обычного стартового сценария, который запускает скрипты **S*** из каталога с соответствующим уровнем выполнения. Другими словами, методы взаимодействия с сервисами времени загрузки не изменились. (Например, для управления ссылками S* и K* все еще используется *chkconfig*.) В обоих случаях ссылка на **/etc/inittab** – это лишь мимолетный кивок в сторону обратной совместимости; в этом файле всего одна строка ‘initdefault’, она задает начальный уровень выполнения. В самом *upstart* нет уровней выполнения, они лишь результат настройки *upstart*, заботящейся об обратной совместимости.

Вообще-то в Ubuntu от идеи уровней выполнения почти отказались, так как их всего два: single-user и 2. Кажется, мы возвращаемся ко временам BSD Unix, где были только уровни выполнения single-user и multi-user, и даже я смог понять, что происходит.

Документация по *upstart* довольно скупа. В частности, в Ubuntu даже нет раздела man, где описывался бы формат файлов событий, хотя такой раздел имеется в Fedora. Неплохое обсуждение есть в блоге Скотта Джеймса Ремнанта [Scott James Remnant] (www.netsplit.com), но как-то странно искать документацию по фундаментальным вопросам в подобных местах. LXF

Это терминал

Заметили, как трудно стало встретить добрую старую командную строку? Раньше вы просто загрузили бы компьютер и сразу ее увидели, но в современных дистрибутивах, чтобы до нее добраться, нужно пройти через три уровня меню. В Vista – четыре. В поисках этой подсказки я теперь начинаю чувствовать себя изгоем, вынужденным мокнуть под дождем на заднем дворе, чтобы выкурить сигарету. Итак, вот мое «впечатление художника» о том, как найти командную строку в Intrepid Ibex.



› Найти командную строку сегодня все сложнее. Сколько прыжков вы осилите?

LINUX FORMAT

Учебники

Наши эксперты помогут вам с любым приложением Linux

В этом выпуске...

60 Успейте больше в Сети
Лучший друг новичков **Энди Ченнел** покажет, как использовать инновационный браузер Flock с социальными сетями вроде Facebook, Bebo и Flickr.

64 Новая система на Eee
Установленный по умолчанию Xandros стал для вас слишком тесен? Следуйте за **Майком Сондерсом** и инсталлируйте себе Mandriva One 2008 или Ubuntu 8.04!

68 Фото в Gimp
Мы познакомились со слоями, текстурами и размыванием – а сегодня **Майкл Дж Хэммел** подберет подходящий способ для скачивания снимков и управления вашими цифровыми фото.

72 Программируем с Rails
В прошлый раз мы набросали простейшую web-галерею для ваших снимков. **Алекс Янг** делает следующий шаг: добавляет Ajax.

76 Пусть ПК заговорит
Чуть-чуть Qt, немного Python – и **Грэм Моррисон** слепит первый (и лучший!) в мире интерфейс к синтезатору речи Festival.

88 Видеопереходы
Ничто так не украшает отснятый материал, как со вкусом подобранные эффекты. **Алексей Маслий** завершает исследования в области Kino и подготавливает фильм к записи на DVD.

80 Плазмоиды
Уже почувствовали, что Plasma – это круто? Научитесь писать виджеты сами и получите первую строку на kde-apps.org – не без помощи **Андрея Боровского**.

92 Соберись, кому говорят!
Даже самые яростные заклинания не заставят программу компилироваться на всем многообразии платформ – а вот CMake это по плечу. Почему? Узнайте у **Андрея Боровского**!

84 Пакеты расширений Scilab
Александр Бикмеев демонстрирует новые возможности победителя Сравнения этого месяца, а заодно предлагает обратить внимание на еще один интересный проект.

96 Функция двух переменных
Вы говорили – мы вас услышали: цикл про R вернулся! Антон Коробейников и **Евгений Балдин** проверят гипотезы однородности и нормальности распределения.

ЕВГЕНИЙ БАЛДИН
Начинал с Агатов. Когда-то даже знал, что такое Робик.

Время пройти квест

Человек начинает жить лишь тогда, когда ему удается преодолеть самого себя.

Альберт Эйнштейн.

Пользователь Живого журнала sadko4u (в миру Владимир Садовников) в своём блоге (sadko4u.livejournal.com/32864.html) разместил занимательный рассказ о том, как он вернул деньги за предустановленную на его личном ноутбуке Windows Vista.

Эта история, содержащая свыше 10 тысяч знаков и три скана, подробно описывает, где и в каком месте на пути избавления от ненужного товара в виде альтернативной операционной системы лежат грабли, и в каком порядке автор их активировал. Очень занимательно. Итог этой прямо-таки фантастической сказки таков: автор вернул $73,90 по курсу ЦБ РФ.

Не менее чем сама история любопытны и комментарии к ней. Хотя подавляющее большинство из них – вполне ожидаемые всхлипы в духе: «Ишь ты, выпендрился. И не лень было такой фигнёй страдать – лучше бы денег заработал», очевидно, что идея отстаивания своих прав достаточно сложно даётся многим из нас. Порадовала и обратная реакция: «Но вы не понимаете! Это же КВЕСТ! Пойди туда – собери такую-то печать – примени бумагу и скиллы, и т.п.» (K48). А ведь действительно, «квест» – самая настоящая игра. Защита своих прав может быть интересной! И боретесь за правое дело, и сложные препятствия имеются, и даже бонус в конце есть, хотя возможность сохранения сильно ограничена. Так почему бы не поиграть?

E.M.Baldin@inp.nsk.su

Совет месяца: Псевдонимы команд

Нам всем знакомо это ужасное чувство: вы набираете **rm ***, палец зависает над клавишей Enter... Тут вы внезапно осознаете, что находитесь не в том каталоге, однако неподвластные уже руки сами завершают короткий, но смертельный жест, и вам остается лишь беспомощно наблюдать, как бывшие минуту назад ценными файлы превращаются в бесполезную кашу из единиц и нулей.

Многие из команд Unix деструктивны по своей природе. *rm* удаляет данные, *cp* или *mv* перезаписывают их без толики колебаний и сожаления. Правда, есть опции, позволяющие подавить опасное поведение – аргументы **-i** или **--intercative** заставят любую из трех команд запрашивать подтверждение на каждый свой шаг, но если б вы могли остановиться и подумать о ключах командной строки, то, наверное, взглянули бы и на каталог. Возможный выход – активировать «безопасный режим» по умолчанию, для этого всего лишь нужно добавить в **/etc/profile** или **~/.bashrc** следующие строки:

```
alias cp='cp -i'
alias mv='mv -i'
alias rm='rm -i'
```

Теперь команды всегда будут вызываться с ключом **-i**, но вы сможете включить тотальное уничтожение, добавив ключ **-f** к вызову утилиты.

Псевдонимы команд не ограничены одним лишь предотвращением файлового Армагеддона – им можно найти и другие, мирные применения. Например, добавить к **ls** или **df** параметр **-h**, чтобы видеть размеры в удобных KB, MB или GB.

Flock: Будем

Весь мир уже давно в Facebook, что дает **Энди Ченнелу** отличный повод изучить web-браузер для общения.

Наш эксперт

Энди Ченнел
Энди делает свои первые шаги в Linux уже шесть лет, а технологиями интересуется со времен Dragon 32.

Firefox прекрасен, но, перефразируя название романа Джанет Уинтерсон [Jeanette Winterson], оранжевый и голубой – не единственный логотип. Полным-полно других браузеров, которые, благодаря модели разработки свободного ПО, основаны на том же коде, что и *Firefox*, но преследуют иные цели. Один из них – *Flock*, базирующийся на технологиях Mozilla и разработанный специально для тех, кто населяет мир сайтов общения вроде Facebook и Flickr и знать не знает о событиях, не попавших в блоги.

Последним релизом *Flock* является 1.2 (бета-версия 2.0, на базе *Firefox 3.0*, вот-вот выйдет), доступный на сайте проекта (**www.flock.com**). В отличие от многих других приложений, он не поставляется с установщиком как таковым; нужно лишь загрузить архив, распаковать его и перенести содержимое в директорию (для подобных случаев в наш домашний каталог в **home** вложена папка **Приложения**). Чтобы запустить *Flock*, можно просто дважды щелкнуть на файле с именем **flock** в каталоге, затем пройти через мастер первого запуска; он предложит взять настройки из *Firefox*, затем отобразит главное окно с двумя уже загруженными вкладками.

Осмотрев меню, вы найдете несколько лучших функций *Firefox* (список недавно закрытых вкладок, дополнения и т.д.), но есть и дополнительные пункты, отсутствующие в стандартном *Firefox*. Вообще-то это верно для любого элемента пользовательского интерфейса *Flock*. Но прежде чем начать их рассматривать, необходимо снабдить приложение некой информацией; в отличие от других браузеров, *Flock* ставит сетевую жизнь пользователей во главу угла всего проекта. Перед тем, как что-либо сделать, уберите стартовое окно, нажав на кнопку, появившуюся у правого края желтой полосы предупреждения в верхней части окна (если вы просто закроете его, оно появится при следующем запуске). Теперь вы должны увидеть динамическую страницу Мой мир [My World], разработанную для отображения самой интересной для вас информации. Сперва она разделена на три части, и пока не добавлено «ничего личного», заполнена стандартным содержимым. Нужно будет немного рассказать ей о себе.

Досье фаната

Есть два способа добавления информации. В центральной колонке страницы Мой мир имеется список социальных сервисов, которые можно присоединить к *Flock*. Чтобы увидеть это в действии, щелкните на ссылке Facebook – вы увидите стандартное приглашение; введите учетные данные, нажмите кнопку Войти [Log in] – и вы уже на обычной странице Facebook. *Flock*, однако, соберет коллекцию новостных лент с этой страницы и отобразит их в боковой панели Люди [People], которая появится у левого края главного окна. При открытии боковой панели Люди на панели инструментов *Flock* подсвечивается иконка Люди. А вернувшись на страницу Мой мир, вы обнаружите, что центральная колонка теперь заполнена информацией с ваших новостных лент Facebook.

Важно запомнить, что боковая панель не статична. Когда ваши друзья по Facebook обновят свой статус, вы узнаете об этом, и вы также увидите «толчки», сообщения, бессмысленные запросы vampire/ninja и даты у самого верха панели. Щелчок на любой из этих иконок перенесет вас на соответствующую страницу. Более того, информация, содержащаяся в элементе каждого друга, также динамична: можете щелкать на их именах, чтобы перейти к профилям; на медиа-кнопке – чтобы просмотреть фотографии; и выберите кнопку Действия [Action], чтобы «потрясти» респондента, отправить сообщение, нарисовать на стене или послать подарок. По сути, *Flock* означает, что вам больше не будет досаждать реклама в Facebook – разве не здорово?

Кроме добавления учетных записей на странице Мой мир, мы можем также воспользоваться боковыми панелями Учетные записи

› Flock открывает это окно при старте. Кнопкой на желтой панели вверху можно пресечь его появление при каждом запуске.

Месяц назад Мы упорядочили свою жизнь с помощью *Tomboy*, *Tasque* и *Thunderbird*.

общаться в web

› Изображения в блоге, электронном письме или в сети размещаются всего за пару щелчков – но убедитесь, что вы не нарушаете авторских прав!

[Accounts] и Сервисы [Services], доступными по щелчку на соответствующей иконке панели инструментов *Flock*. По мере добавления новых учетных записей они будут появляться здесь, для облегчения доступа. Можно, например, добавить учетную запись Flickr, щелкнув на ссылке и введя имя и пароль на открывшейся главной странице Flickr.com, а затем проверить, появились ли наши контакты в боковой панели. Facebook и Flickr теперь имеют свои собственные вкладки в боковой панели Люди.

Если вы работаете под учетной записью в момент попытки подключиться к ней через *Flock*, может понадобиться сначала выйти, чтобы приложение опробовало реквизиты. Однако *Flock* по умолчанию запоминает ваши данные, и достаточно ввести их лишь однажды, если только не щелкать на ссылках Забыть учетную запись [Forget Account] или Выйти [Log Out] в менеджере учетных записей.

Медиа-панель

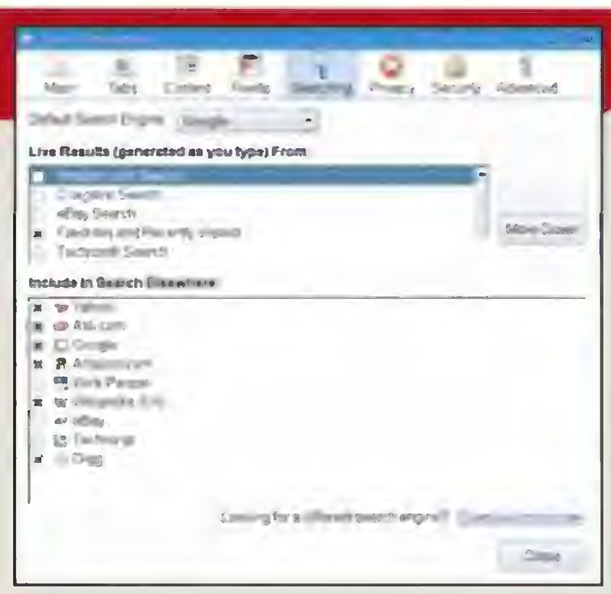
Щелчок на ссылке Медиа [Media] в любой учетной записи запустит другую большую часть *Flock*, Медиа-панель. Это полоса миниатюр, разбросанных по верхнему краю окна браузера. Если вы войдете под учетной записью Flickr или Picasa, изображения на Медиа-панели загрузятся из нее, а если вы щелкнете на медиа-ссылке под чьим-то именем, то увидите его изображения, взятые с Flickr, Picasa или Facebook. Задержите над миниатюрой мышь, чтобы увидеть картинку большего размера; щелкните на изображении, чтобы узнать, с какой страницы оно пришло; или щелкните на стрелке, направленной вправо, для вывода дополнительных опций, включая копирование URL с различными размерами изображений, добавление картинок в блог или пересылку друзьям по электронной почте.

Имеется много других способов публикации информации во *Flock*. Например, настроив учетную запись блога, вы сможете отсылать сообщения на любой сайт, просто открыв редактор щелчком на иконке с изображением пера. Кроме хороших средств работы с текстом, редактор блогов *Flock* содержит опции для сохранения локальных черновиков, размещения в нескольких блогах и переноса клипов мышью с других сайтов для размещения в сообщениях. Для фотографов приложение содержит очень простой инструмент отсылки изображений в учетные записи *Flickr*, Picasa или Facebook. Чтобы открыть его, выберите иконку загрузки (это стрелка на панели инструментов *Flock*, указывающая вверх), укажите свой сервис и перенесите фотографии

Поиск

В отличие от *Firefox*, окно поиска во *Flock* по умолчанию использует Yahoo, а не Google. Если вы хотите сменить поисковик, вам потребуется открыть диалог Правка > Настройки [Edit > Preferences] и перейти на вкладку Поиск [Search]. Верхний раздел задает поисковик по умолчанию [Default], то есть тот, что указан рядом с адресной строкой. Второй раздел – для настройки предварительных результатов, появляющихся под строкой Поиск по мере набора запроса; вы можете повысить производительность браузера, отключив эту опцию.

Последний раздел – набор поисковиков, появляющийся под опцией Искать в других [Search Elsewhere]. Если, по-вашему, чего-то не хватает, щелкните на ссылке Загрузить дополнительно сейчас [Download More Now] или выберите более простой метод и перейдите на сайт, при помощи которого хотите искать. *Flock* отметит, что сайт имеет опцию поиска, и предложит добавить его в список доступных поисковиков.

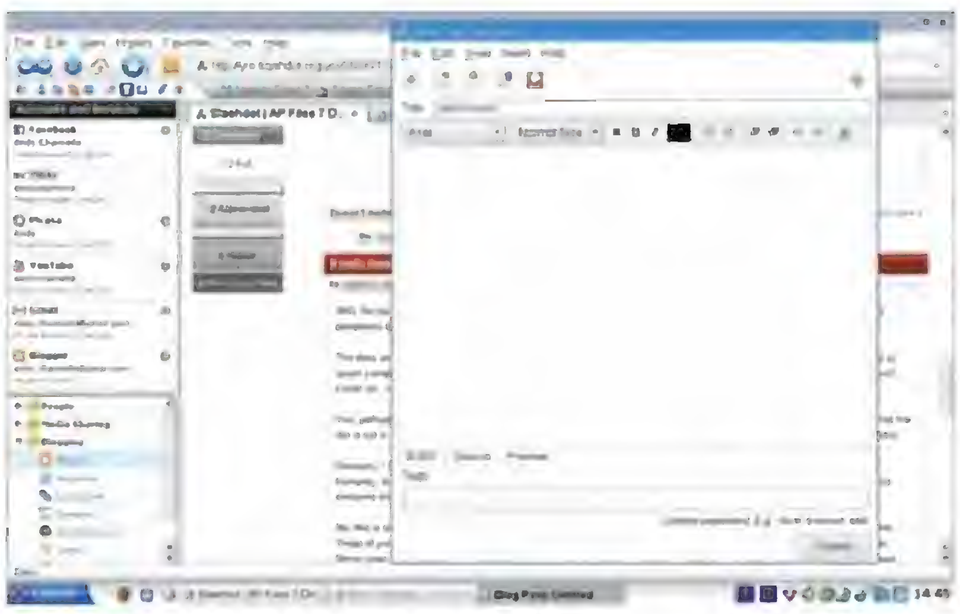


› *Flock* обладает всеми основными опциями поиска, и всегда можно добавить еще.

из любого места файловой системы прямо в вашу учетную запись. В загрузчике также есть место для добавления тэгов, описаний и опций приватности, и, как и для других частей системы, вы должны авторизоваться в своем сервисе перед началом загрузки.

Другая приятная возможность – перенос ссылок, видео или изображений любым вашим друзьям в Facebook. Убедитесь, что в боковой панели Люди выбран Facebook, затем перенесите какие-либо медиафайлы на чье-нибудь имя, и *Flock* откроет стандартную страницу обмена с готовыми для заполнения именем, темой и ссылкой.

Многие пользователи полагают, что компьютеры и связь придуманы исключительно ради общения в web. Вы можете добавить к *Firefox* множество дополнений, чтобы получить все эти функции в одном флаконе, но *Flock* значительно упрощает решение данной задачи. Единственная проблема в том, что если вы откроете боковую панель Люди, включите Медиа-панель и окно блога или загрузчик фотографий, останется очень мало места для самого содержимого; но если работа с социальными сетями и есть это содержимое, то он идеален. LXF



› *Flock* имеет встроенный редактор блогов, прекрасно работающий с *WordPress*, *TypePad* и *Blogger*. Работал бы он с Tums...

Скорая помощь

Если вы пользуетесь общественным компьютером, не забудьте выйти со всех сайтов перед тем, как покинуть свое место.

Скорая помощь

Настройте доступ к вашей учетной записи web-почты через иконку электронной почты на панели инструментов *Flock*.

Google Desktop:

Кроме доступа к огромному диапазону мелких настольных приложений, Google Desktop предлагает опции поиска. Позвольте **Энди Ченнелу** представить вам…

После долгого вынашивания, Google Desktop наконец-то достиг функциональной эквивалентности в Linux, Mac OS X и Windows, то есть пользователи Linux могут не только выполнять поиск на рабочем столе/в сети, но также загружать и устанавливать разнообразные настольные гаджеты.

ПО можно скачать с сайта Google http://desktop.google.com/en/linux/download.html, и оно доступно в форматах RPM и Deb (для Red Hat и дистрибутивов на базе Debian, соответственно) для архитектур x86 и x86-64. Объем загрузки меньше 8 МБ, что не заклинит ваш канал, а после скачивания файл можно открыть в менеджере пакетов вашего дистрибутива. Если не получилось, перейдите в каталог с ним и выполните двойной щелчок: видимо, потребуется ввести пароль администратора-root, чтобы приложение можно было установить.

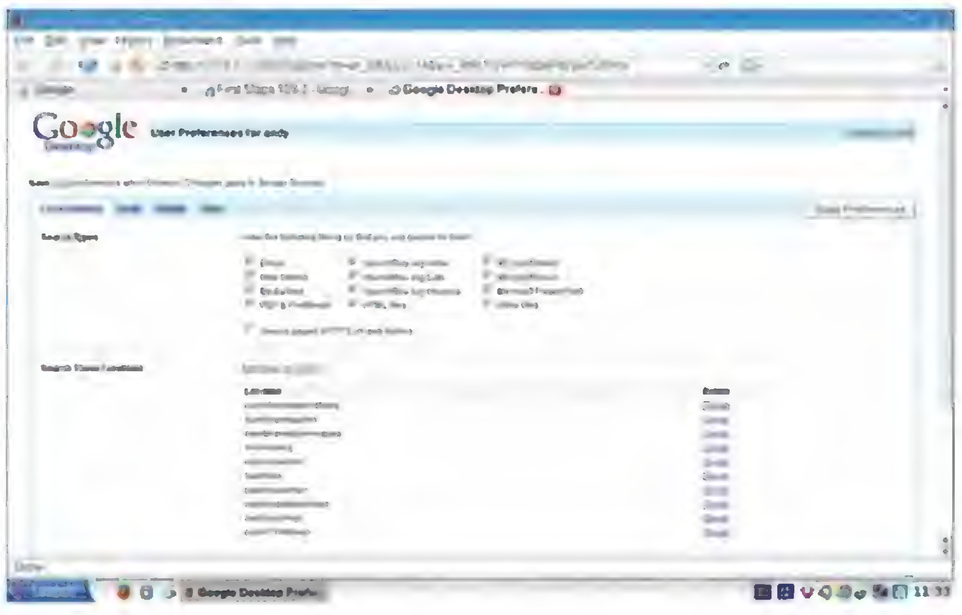
Скорая помощь

Мы засекли несколько проблем Google Desktop в интеграции с *Firefox 3*, и если вы используете Firefox для чего-то важного, оставайтесь пока с версией 2.

На нашей тестовой системе Gnome (Ubuntu Hardy Heron) программа добавила пару новых иконок в меню Приложения > Другие, а в KDE (Xandros Desktop 4) появилось новое меню Google Desktop. Два элемента там – это запуск самого приложения и конфигуратора; с него мы и начнем. Выполните Приложения > Другие/Google Desktop > Параметры Google Desktop для запуска мастера настройки. Вас спросят, желаете ли вы сделать Google вашей домашней страницей и хотите ли отсылать отчеты об ошибках и анонимную статистику использования. Когда вы примете условия лицензии (ведь вы уверены, что Google – это не зло, правда?), в область уведомления будет добавлена новая иконка (справа вверху в Ubuntu, справа внизу в KDE; правый щелчок на ней позволит вам позднее указать некоторые другие параметры), а также запустится еще один экземпляр *Firefox* для настройки приложения.

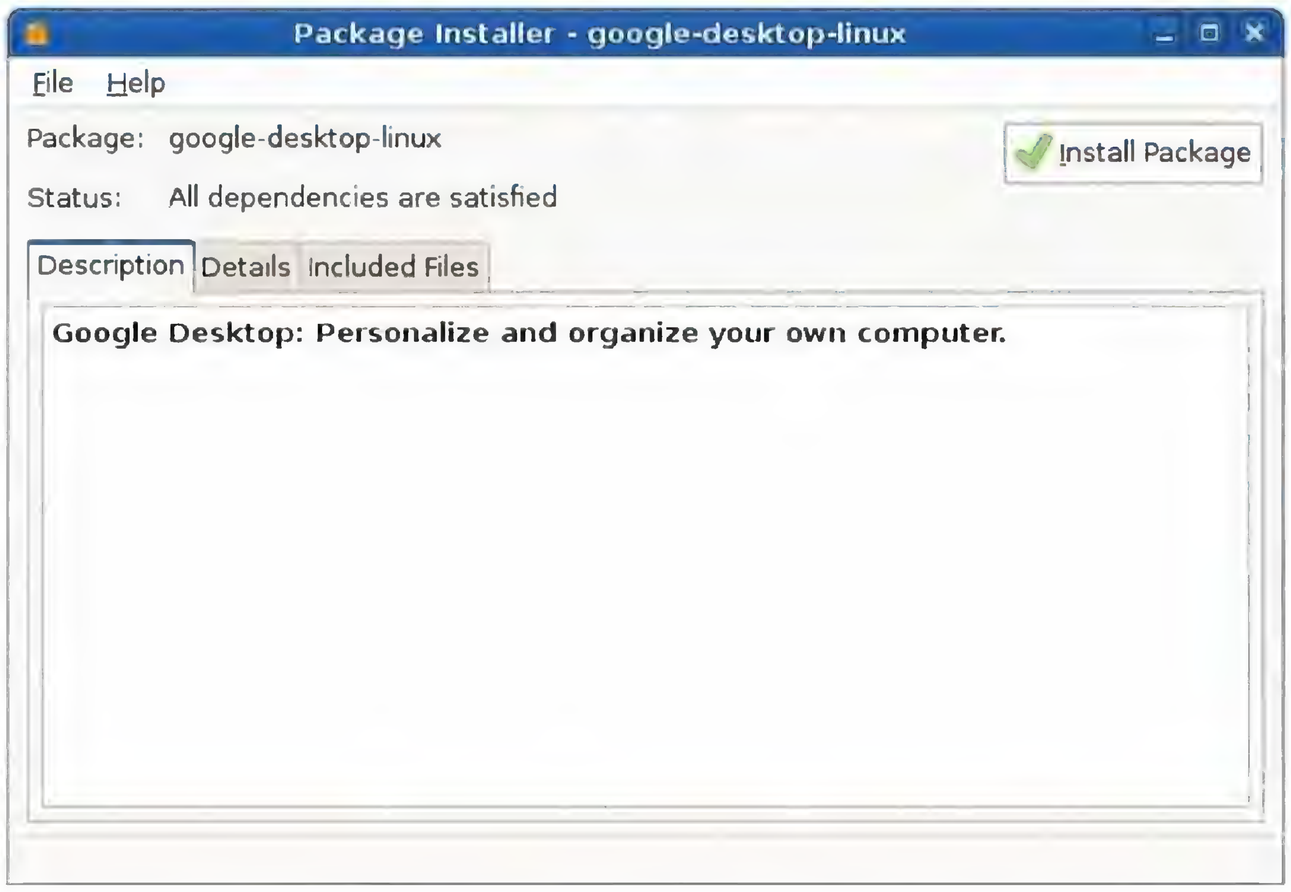
В этом окне имеется четыре вкладки, описывающие локальный поиск, Gmail, Отображение [Dislpay] и Другие [Others]. Будьте с ними очень осторожны: можно подорвать собственную безопасность, позволив приложению индексировать то, что должно быть защищено.

Начнем с построения локального индекса. При этом ваша система будет полностью просмотрена, а расположение ваших файлов занесено в журнал, чтобы потом их легко было найти. В верхней части окна отображаются типы файлов, которые индексирует Google Desktop, и они включают все, от PDF и документов *OpenOffice.org* до файлов *Microsoft Word* и сохраненных локально HTML-страниц. По умолчанию программа также индексирует вашу историю web – так что можно искать недавно посещенные страницы – и электронные сообщения вашего стандартного почтового клиента. Понятно, что при использовании электронной почты для любовной переписки или промышленного шпионажа любой формы вы можете пожелать запретить приложению индексировать ваши письма.



› Настройки Google Desktop доступны на web-странице в стандартном браузере.

› Установочный пакет занимает всего лишь 8 МБ (почти как *Firefox*) и добавляется при помощи вашего обычного менеджера пакетов.



Второй раздел сообщает программе, где искать. Мы можем добавить новые места, щелкнув на ссылке вверху этого списка и добавив соответствующий путь. По умолчанию в этом разделе перечисляются местоположения стандартных приложений (Google Desktop можно использовать и для их запуска по команде) и ваши домашние каталоги. В третьем разделе можно исключить из индексирования файлы, каталоги и URL. URL добавляются копированием из адресной строки браузера, а конкретные файлы и каталоги исключаются при помощи ссылки в левой части текстовой области.

В нижней части окна имеется опция исключения из индекса файлов, которые были удалены. Щелкните на ней, если вы не хотите, чтобы удаленные файлы засоряли результаты поиска. Настроив все как следует, нажмите Сохранить настройки [Save Preferences], и мы можем двигаться дальше.

Вкладка Gmail позволяет приложению индексировать существующие учетные записи Gmail: просто введите учетную информацию и нажмите Сохранить настройки [Save preferences], чтобы добавить ее. Вкладка Отображение [Display] настраивает внешний вид результатов и способ активирования строки поиска. Мы оставим все как есть, сменим только поиск с Сети [Web] на Рабочий стол [Desktop].

Наконец, на вкладке Другие можно выключить или отключить опцию отправки статистики в Google и почитать о его политике конфиденциальности. Пока вы делаете это, приложение проиндексирует вашу систему, на что требуется время, особенно если надо поработать с большим объемом данных.

изучаем гаджеты

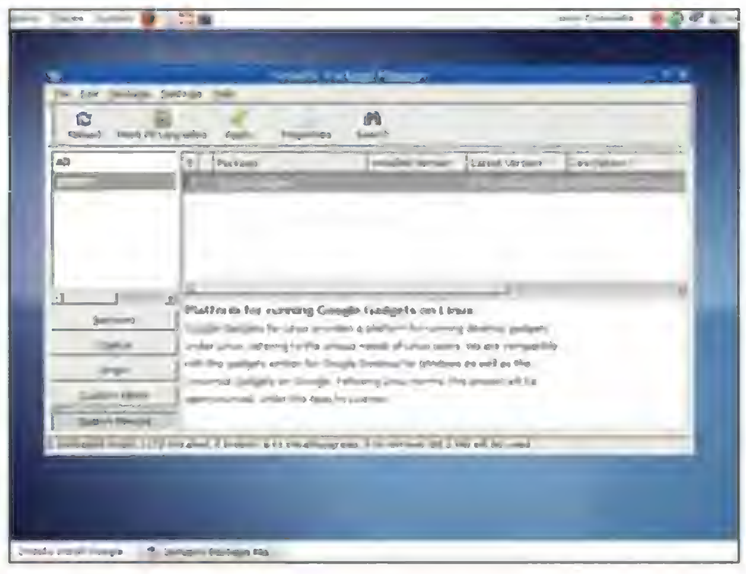
Шаг за шагом: Управляем гаджетами

1 Открываем строку поиска

Чтобы начать, быстро нажмите клавишу Ctrl два раза (можно также назначить другую комбинацию). В центре экрана появится полупрозрачная строка ввода. Как только вы начнете набор, под ней появятся результаты из Интернета или с вашей локальной машины (зависит от настроек).

2 Просмотр результатов

Выберите элемент списка для открытия файла, электронного письма или web-страницы в соответствующем приложении. Щелчок вне строки поиска/результатов приведет к их исчезновению, но при двойном нажатии Ctrl они появятся в том же виде, в котором вы их оставили. Это упрощает просмотр нескольких результатов без повторного ввода искомого. Чтобы увидеть все, выберите внизу списка Показать все результаты в браузере [Show all results in browser].



3 Добавляем репозиторий

Когда вы будете читать эти строки, для многих дистрибутивов уже появятся двоичные гаджеты Google. Пользователи Ubuntu могут добавить **deb http://ppa.launchpad.net/googlegadgets/ubuntu hardy main** в Synaptic (выполните Настройки > Репозитории > Сторонние [Settings > Repositories > Third Party] и нажмите Добавить [Add] для добавления репозитория), Нажмите Обновить [Reload], затем поищите Google Gadgets и установите дополнительные приложения.

4 Вперед, гаджет!

Для первого запуска приложения-гаджета нажмите Alt+F2 (откроется диалог Выполнить [Run]), а затем введите **ggl-gtk -s -bg**. Это добавит еще одну иконку в область уведомлений и создаст боковую панель для отображения гаджетов у правого края экрана. Чтобы добавить гаджет, щелкните правой кнопкой на этой иконке в области уведомлений и выберите Добавить [Add].

5 На рабочем столе

Можно запустить гаджеты и вне боковой панели, набрав в диалоге Выполнить **ggl-gtk**. Проблема здесь в том, что гаджеты пока имеют в качестве фона черный квадрат, и будут ужасно выделяться на цветных обоях. Можно также запускать боковую панель гаджетов при загрузке, создав ссылку на команду на вашем рабочем столе и затем добавив ее в автозапуск KDE или Gnome.

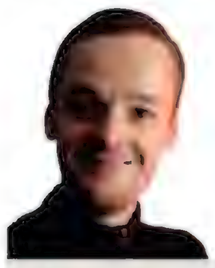
6 Язык англов

Если вы не меняли язык в настройках на 'English', то будут доступны не все гаджеты. Отсортируйте доступные гаджеты и нажмите кнопку Добавить, когда увидите что-то классное. Гаджет добавится на боковую панель, и его можно будет перетащить. Другие гаджеты передвинутся, отобразив ваши изменения. LXF

Через месяц Альтернативы *Browser Sync* от Google: *Opera 9.5* и *Weave* от Mozilla.

Eee PC: Смена

Часть 3: Недовольны ограничениями встроенного дистрибутива Xandros? **Майк Сондерс** научит, как сменить систему на вашем микроноутбуке.



Наш эксперт

Майк Сондерс прикупил Eee PC еще в ноябре, и сделал все мыслимые и немыслимые настройки. Он носится со своим ноутбуком как с маленьким пластмассовым ребеночком.

Мы подошли к последней части нашего учебника о том, как добиться большего от Asus Eee PC, но самую потеху приберегли на конец. Месяц назад мы рассматривали установку добавочного дистрибутива на SD-карту, создав возможность выбирать между загрузкой Xandros и облегченным Breeezy. Сегодня мы доведем дело до конца и поставим полный дистрибутив на главный накопитель Eee.

Спросите, зачем вам это надо? Конечно, стандартный Xandros великолепен для повседневных задач: быстро загружается, содержит все важные приложения, позволяет добавлять новые. Но у него есть и недостатки, которые можно устранить сменой дистрибутива:

» **Безопасность** Asus всегда тянет резину с выпуском обновлений безопасности для Eee, что сильно раздражает некоторых пользователей. Лишь пользуясь современным, регулярно обновляемым дистрибутивом, можно быть уверенным, что вам ничего не угрожает.

» **Универсальность** Репозитории Xandros полны приложениями, но их не столько, сколько имеется у Ubuntu, и есть риск перезаписи пакетов, специфичных для Eee (например, ядра) типовыми пакетами Xandros'а. Имея же полный универсальный дистрибутив, вы сможете брать больше программ и не опасаться, что один пакет испортит другой.

» **Привычка** Может быть, вам просто удобнее работать с другим дистрибутивом, где вы разобрались с файловой иерархией и утилитами настройки. Поменяв систему, вы сможете воссоздать ваш настольный компьютер на Eee.

На данном уроке мы рассмотрим два дистрибутива: Mandriva и Ubuntu. Первый, в своем воплощении 2008 Spring, идеально подходит для Eee, тогда как добиться гладкой работы Ubuntu немного сложнее. Чтобы выполнить наши инструкции, Eee потребуется 4 ГБ на внутренней флэш-памяти (модели с 2 ГБ могут работать с меньшими дистрибутивами, но не с Mandriva или Ubuntu: тем понадобится серьезная настройка).

Часть 1 Установка Mandriva

Для инсталляции нового дистрибутива на Eee вам понадобится внешний CD/DVD-привод. Установка системы с USB-брелка тоже возможна, но зачастую это очень тонкая и сложная работа, требующая большой возни с утилитой *fdisk*, создающей разделы диска. А чтобы начать инсталляцию с внешнего оптического накопителя, достаточно просто вставить CD. Если у вас нет такого привода и негде его занять, придется купить.

Переживать за свой бюджет тут особо нечего: в онлайн-магазинах они стоят меньше 2000 рублей, а если вам повезет найти чистый CD-ROM (без DVD) – то еще дешевле. Какое бы из стандартных внешних USB-устройств без лишних наворотов вы ни взяли, оно отлично заработает с Eee. Кто не уверен в выборе, расспросите сторонников Eee на дружелюбном форуме http://forum.eeeuser.com или www.linuxforum.ru.

Учтите, что двигатель привода CD/DVD способен скорехонько посадить батарею; будет разумным, присоединив привод к Eee и готовясь начать установку дистрибутива, подключить мини-ноутбук к сети питания. Куда как неприятно проделать 70% работы, увидеть зловещее мигание лампочки, забегать в поисках адаптера питания и вернуться к погасшему ноутбуку. При подключенном и работающем внешнем приводе батарея Eee протянет в лучшем случае час или около того, вместо привычных трех.

Это просто работает!

Последний выпуск Mandriva, Spring 2008, полностью поддерживает оборудование Eee PC. На Eee отлично работают беспроводная сеть, управление питанием и экранные функции, а после основной установки не придется тратить время на поиски файлов конфигурации и дополнительных пакетов. На диске **LXF**DVD в разделе Дистрибутивы/Mandriva вы найдете ISO-образ Live-дистрибутива **mandriva-one-2008-spring.iso**, не требующего установки. Запишите его на CD-R диск программой K3b, воспользовавшись меню Инструменты > Записать ISO-образ [Tools > Write ISO Image]. Для загрузки и установки Mandriva на ваш Eee следуйте инструкциям справа.

» **Месяц назад** Мы установили новый дистрибутив на SD-карту и настроили резервирование.

LINUX FORMAT
На DVD
» Mandriva One 2008 Spring
» MadWifi r2756

дистрибутива

Шаг за шагом: Установка Mandriva на Eee

1 Запуск

Подключите ваш внешний USB-привод к Eee, вставьте Mandriva CD-R и нажмите Esc на экране загрузки. Теперь можно стартовать с внешнего привода CD/DVD, а не внутреннего флэш-накопителя.

2 Загрузка

Mandriva загрузится. Вам зададут вопросы про язык и настройки клавиатуры, и нужны ли вам 3D-эффекты рабочего стола – отключите их, если гонитесь за производительностью.

3 Установка

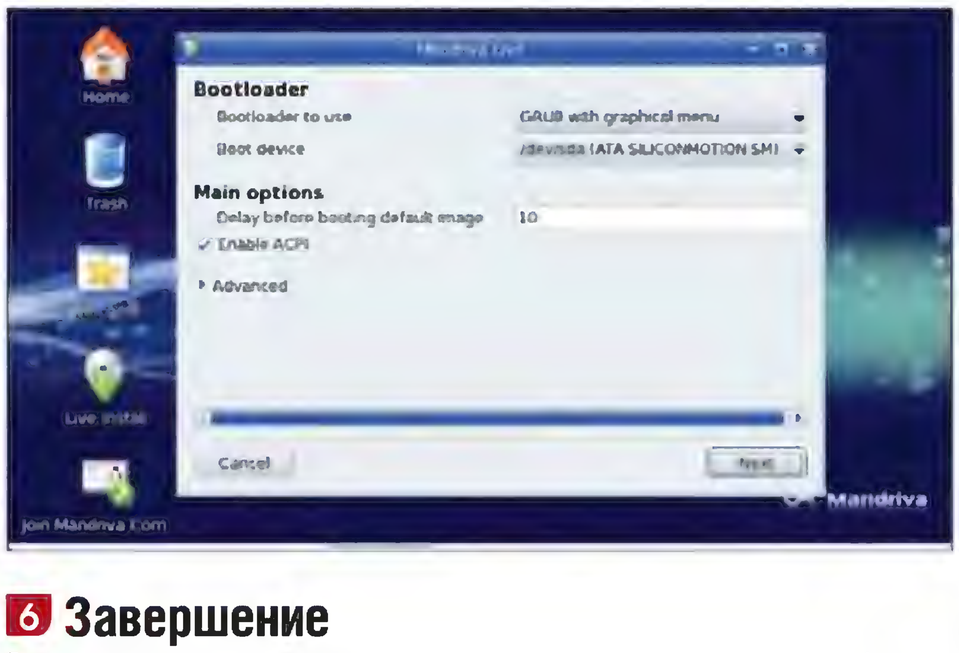
Для постоянной установки Mandriva дважды щелкните по значку «Live install» на рабочем столе. Учтите, что на внутреннем флэш-накопителе Eee все будет удалено, так что предварительно сделайте резервные копии!

4 Создание разделов

Самый простой способ – предоставить Mandriva управляться самой, так что выберите «Erase And Use Entire Disk», задав в качестве устройства **sda** (внутренний накопитель).

5 Ожидание

Mandriva пойдет копировать свои файлы на Eee. На нашей машине с внешним CD/DVD скорости 24x процесс занял примерно 10 минут. Если ваш привод медленнее, времени потребуется больше.



6 Завершение

Вас попросят подтвердить несколько установок для загрузчика (просто примите параметры по умолчанию). Нажмите Далее [Next], затем Готово [Finish], затем щелкните по кнопке Меню (слева внизу), завершите сессию и выключите компьютер.

Теперь дистрибутив ваш; нажмите кнопку питания и включите Eee. При первой загрузке вам будет задано с десяток вопросов по настройке, первый из которых касается сети. Вы можете настроить и проводное (Ethernet), и беспроводное соединение. В последнем случае система предложит список Wi-Fi-сетей, к которым есть возможность подключиться.

Затем вам нужно будет задать пароль для пользователя root (администратора), а также создать учетную запись для обычного пользователя. Не забывайте информацию, которую вы здесь введете! После этого система предложит вам присоединиться к сообществу Mandriva, но если вас это не интересует, жмите Отклонить [Decline]. Наконец, вы увидите экран входа в систему: введите заданные вами выше имя пользователя и пароль – и вот вы на рабочем столе KDE.

Использование Mandriva

Mandriva – сама простота: для просмотра установленных программ нажмите кнопку Меню в левом нижнем углу; для настройки системы или установки нового оборудования щелкните по значку Настроить компьютер [Configure Your Computer] на нижней панели (сразу слева от иконки *Firefox*). Обратите внимание, что превосходно работают все операции, контролируемые клавишей Fn (например, яркость экрана или громкость), и в целом установка на Eee поразительно гладкая. С вопросами или проблемами, или просто при желании пообщаться с пользователями, установившими Mandriva на свои Eee, обращайтесь на сайт **http://forum.eeeuser.com**, где есть отдельная ветка, посвященная этому дистрибутиву.

Центр управления Mandriva заведует аппаратной частью и устанавливает дополнительные пакеты.

Часть 2 Установка Ubuntu

Ubuntu 8.04 работает на Eee хорошо, хотя его поддержка оборудования не столь полная , как у Mandriva, и после установки придется немного поработать. К счастью, действия по установке дополнительных драйверов и настройке интерфейса для экрана с низким разрешением можно автоматизировать готовыми скриптами. Если у вас есть экземпляр **LXF107**, можете установить Ubuntu с LXFDVD, если же нет, или ваш внешний привод только CD, придется скачать ISO-образ Ubuntu 8.04 с **www.ubuntu.com** и записать его на CD-R.

Скорая помощь

Обратите внимание, что полная смена дистрибутива удаляет с Eee скрытый раздел восстановления. Если вы хотите вернуть Xandros и у вас нет резервной копии, загрузите ваш Eee с поставляемого с ним DVD.

Вставьте диск в ваш внешний привод, подсоедините его к Eee и включите ноутбук. Попав на загрузочный экран Asus'a, жмите на Esc, пока не попадете в загрузочное меню. Здесь вы можете выбрать загрузку с внешнего привода (а не с внутреннего флэш-накопителя) с помощью клавиш курсора. После этого вы увидите загрузочное меню Ubuntu; оно позволит вам запустить дистрибутив в режиме Live (работы непосредственно с CD/DVD) или перейти прямо к установке. Мы воспользуемся последней опцией.

Разомните пальчики

Загрузка у Ubuntu не самая быстрая – один установщик займет пару минут, так что не паникуйте, если вам покажется, что все зависло. Попав на экран инсталлятора, следуйте указаниям, как при обычной установке Ubuntu. На этапе создания разделов выберите Использовать весь диск [Use entire disk] для привода sda. Естественно, это удалит все данные на Eee.

После установки, загрузив Ubuntu с внутреннего накопителя, вы заметите, что разрешение экрана выставлено правильно, звук работает, но кое-чего не хватает – главное, отсутствуют поддержка беспроводного соединения и экранные сообщения для комбинаций клавиш с Fn (изменение громкости и пр.).

Чтобы поправить дело, подключите ваш Eee к проводной сети (Ethernet). После этого откройте окно с командной строкой (Applications > Accessories > Terminal) и введите следующее:

```
wget http://eee.ricey.co.uk/files/eee/RiceeeyTweak.sh
sh RiceeeyTweak.sh
```

Скачается и выполнится скрипт, настраивающий установку Ubuntu по многим пунктам. Вы получите: более мелкие шрифты; лучшее управление питанием; экранные подсказки при изменении громкости и яркости; и поддержку Wi-Fi. Также появится утилита рабочего стола под названием «Overclock», которая разгоняет CPU до 900 МГц (по умолчанию он работает на 630 МГц). Это даст выигрыш в производительности, но ценой большего разогрева и меньшего срока работы батареи.

Скрипт настройки Eee от Ricey обеспечит полную поддержку оборудования и подгонит шрифты к маленькому экрану.

Все готово!

Итак, у вас теперь полный дистрибутив, какой уж вы там выбрали – Mandriva или Ubuntu. Ваш Eee стал мощнее и способнее, и вы можете получать регулярные обновления. Если вы выполняли урок прошлого месяца, то сделали резервную копию установки Xandros, и в случае необходимости сможете вернуться обратно за считанные минуты.

Насладитесь новым дистрибутивом и черкните пару строк о ваших достижениях на форумы сайта **LXF**, **www.linuxforum.ru**. Пишите, если вы нашли трюк, делающий эти дистрибутивы еще лучше, или даже если вы попробовали на вашем Eee совсем другой дистрибутив!

Часть 3 Прочие дистрибутивы

Из крупных дистрибутивов на Eee сейчас лучше всего поддерживаются Mandriva и Ubuntu, но если у вас иные предпочтения, вам есть что попробовать. В разработке находятся многие побочные проекты, где хакеры – владельцы Eee берут свои любимые разновидности Linux и настраивают их для эффективной работы с оборудованием Eee и на крохотном экране. Взглянем на наиболее заметные альтернативы для Ubuntu и Mandriva на Eee…

EeeDora

Основан на Fedora

Сайт www.complexvalues.com

Ежу понятно, что EeeDora – дистрибутив, поддерживаемый сообществом Red Hat. Он доступен в двух ипостасях: загрузочного ISO-образа, который можно записать на CD-R, и Zip-файла, распаковываемого на USB-брелок. Как упоминалось в начале этого учебника, подготовка дистрибутива к запуску с USB-брелка часто выливается в сложную процедуру, но, к счастью, в Fedora есть утилита, которую можно использовать для преобразования ISO-образа в дистрибутив, загружаемый с USB. Правда, нужно знать узел устройства для раздела USB-брелка – обратитесь к документации на сайте EeeDora за полным списком инструкций по установке.

EeePCLinuxOS

Основан на PCLinuxOS

Сайт www.eeepclinuxos.com

Как вы, наверное, уже слышали, PCLinuxOS – стремительно восходящая звезда в мире дистрибутивов, и каждый день он завоевывает новых поклонников. Сочетание приятной темы рабочего стола, дружелюбных инструментов настройки и происхождение от Mandriva делает его хорошим выбором. Эта специальная версия для Eee находится только на стадии бета 3-1, но уже выглядит впечатляюще, вмещая яркие эффекты рабочего стола, сделавшие PCLinuxOS столь узнаваемым (*см. справа внизу*).

Можете попробовать две версии EeePCLinuxOS: полный ISO-образ (700 МБ) включающий (в сжатом виде) более 2 ГБ приложений, или меньшую версию Base, вообще без программ. Последняя предназначена для опытных пользователей, которые точно знают, что они хотят добавить после установки, и не растеряются, увидев голый рабочий стол.

Один разработчик EeePCLinuxOS работает над специальным интерфейсом для дистрибутива, имитирующем Easy Mode от Xandros. Он заменит обычное меню Start и панель задач большими толстыми кнопками, что идеально для новичков и детей, которым надо запускать только несколько приложений – изначальный целевой рынок Eee.

Заставим работать беспроводную сеть

Если вы выбрали для установки дистрибутив, не имеющий специальной поддержки Eee, основным препятствием для вас будет беспроводная сеть. Новейшие дистрибутивы могут поддерживать чип Atheros Wi-Fi в Eee; если вы только-только установили новый дистрибутив и обнаружили, что не можете попасть в онлайн по воздуху (а лишь по Ethernet), установите драйверы MadWifi. Вы можете взять их в разделе **Система/MadWifi** на LXFDVD.

Возьмите файл **madwifi-ng-r2756+ar5007.tar.gz** и сохраните его в домашний каталог вашего Eee. Для компиляции драйвера нужна среда сборки (*GCC*, *make* и Ко), установленная в вашем дистрибутиве. Введите такие команды:

```
tar zxf madwifi-nr-r3366+ar5007.tar.gz
cd madwifi-ng-r3366+ar5007
make clean
make
sudo make install
```

(Если **sudo** откажется работать, попробуйте ввести просто **su**, чтобы переключиться в режим root'а). В разделе **Справка** нашего DVD вы найдете подробную информацию об учетных записях пользователей. Закончив, перезагрузите Eee, и беспроводное соединение должно включиться.

Ubuntu Eee

Основан на Ubuntu

Сайт www.ubuntu-eee.com

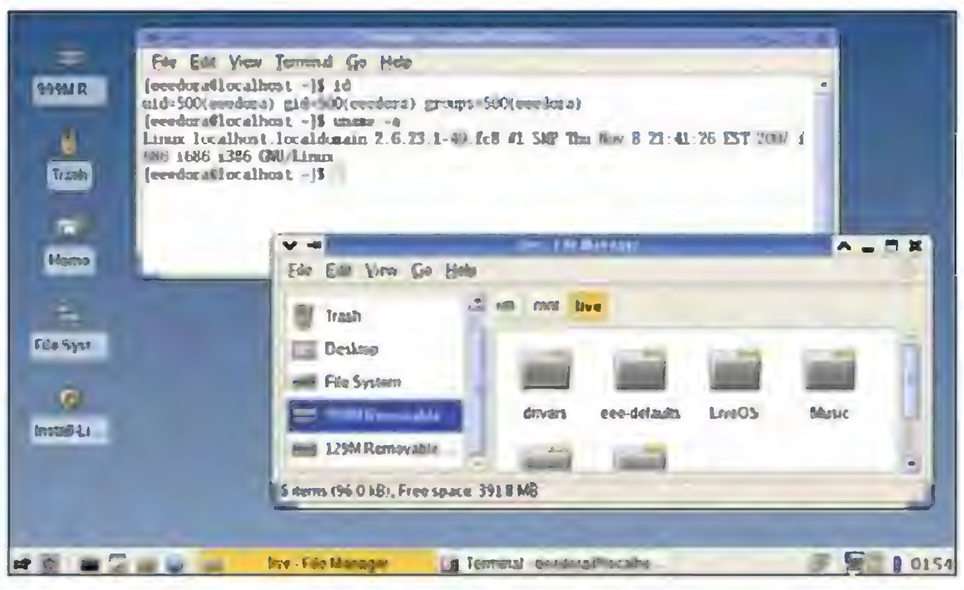
Проекты для объединения Ubuntu с оборудованием Eee разрабатываются вовсю, но наиболее обещающий сейчас тот, что изобретательно назван Ubuntu Eee. Он основан на Ubuntu 8.04 и подгружает необходимые драйверы и настройки, чтобы дистрибутив запускался гладко, не хуже скрипта **RiceeeyTweak.sh** с предыдущей страницы. Ubuntu Eee вышел как раз когда мы заканчивали этот учебник, и у нас не было времени его протестировать. Кто рискнет попробовать – дайте нам знать, что у вас получилось.

Zeee

Основан на Zenwalk

Сайт http://tinyurl.com/6ldtu9

Zenwalk – один из самых резвых дистрибутивов, а значит, отличный кандидат для установки на Eee. Его разработка шла неплохо примерно до начала лета, а затем наступил спад. Но это очень обещающий и легкий дистрибутив, и если вы набили руку на командной строке и хотите вдохнуть новые силы в проект, просмотрите форум по ссылке выше (обратите особое внимание на посты основателя проекта Alisou). LXF



› **EeeDora основан на Fedora 8 и использует рабочий стол *Xfce*, чтобы сэкономить память и добиться большего от скромного процессора Celeron.**

› **EeePCLinuxOS скоро заимеет новый дружелюбный GUI, подражающий Easy Mode в Xandros.**

Gimp: Работаем с

В *Gimp* столько всяких функций, что многим хватает лишь малой их толики. **Майкл Дж. Хэммел** подключил свою камеру к ПК и от*Gimp*ил пару снимков.

Наш эксперт

Майкл Дж. Хэммел является участником проекта *Gimp* и автором трех книг по данной теме, включая последнюю *The Artist's Guide to Gimp Effects* [Введение в эффекты Gimp для художников].

Эта статья посвящена легендарному приложению из мира свободного графического ПО – GNU Image Manipulating Program, более известному как *Gimp*. *Gimp* – редактор изображений, и это означает (кроме всего прочего), что он хорошо обрабатывает фотографии. При помощи *Gimp* можно подстраивать цвета или удалять их вовсе, накладывать текст и даже замещать фон. Но прежде чем что-то делать, следует найти способ передачи фотографий в программу.

До цифровых фотокамер мы героически (хоть и не всегда успешно) сражались с планшетными сканерами. Эти устройства не очень хорошо поддерживались в Linux (невзирая на самоотверженные усилия команды Sane), и выдаваемый ими результат сильно зависел от смеси качества ПО с качеством фотопленки, не говоря уже о состоянии стекла и оптики сканера. При стольких потенциальных источниках помех, не замедлил появиться вариант получше.

Ликбез по цифровым камерам

Цифровые камеры предоставляют более простой метод получения изображений и передачи их в популярные цифровые инструменты: в компьютер, в программы, используемые для редактирования фотографий, и в различные устройства хранения. Больше не нужно заботиться о преобразовании из одного формата (пленка) в другой (цифровой файл) – камера делает это за нас. Теперь мы работаем с цифровыми файлами напрямую. Сегодня это просто жесткий диск или USB-брелок. Жизнь продолжает становиться проще.

Но даже с цифровыми камерами все еще приходится учиться. Нужно знать, как вытащить изображения, как хранить их, какой способ работы с ними лучше и какие приложения тут могут понадобиться.

Данная статья нацелена в первую очередь на рядового потребителя. Профессиональные пользователи цифровых камер могут свободно обращаться ко мне по специфическим вопросам практического применения *Gimp* для обработки изображений. Большинство цифровых камер бытового уровня имеют несколько общих черт:

- Они поддерживают соотношение сторон 4/3 или 3/2.
- Они предоставляют один из двух (или оба) метода передачи файлов.
- Они сохраняют фотографии внутри, чтобы вы могли вытащить их позднее.

Большинство современных цифровых камер обеспечивает весьма высокое разрешение. Число мегапикселей вместе с соотношением сторон определяют реальный размер снимков. В случае 4/3 для камеры с 8 МПикс фотографии получаются размера приблизительно 3456x2304 точек. При разрешении печати 250 dpi это даст 13x9 дюймов [~33x23 см] – намного больше, чем стандартные для Британии отпечатки 3x5 дюймов, которые вы заказывали в фотолавочке местного супермаркета. Соотношение сторон легко изменяется в *Gimp* при помощи инструмента Кадрирование, так что вариант, выдаваемый вашей камерой, особого значения не имеет.

Возможные средства хранения охватывают диапазон от мультимедиа-карт до дисков, но потребительские камеры тяготеют к первым, ибо они дешевле. Это подразумевает память, основанную на флэш, такую как карты Secure Digital и Compact Flash, а также USB-диски.

Если и ваша камера, и компьютер поддерживают мультимедиа-карты, то для получения доступа к фотографиям можно просто подключить носитель к вашей системе Linux. Если же ваша камера или компьютер их не поддерживают, то у вас на камере, вероятно, есть гнездо USB для доступа к фотографиям. При наличии такового у вас будет по крайней мере один из двух вариантов передачи снимков на компьютер.

Первый вариант называется PTP, Picture Transfer Protocol [Протокол передачи изображений]. PTP определяет, как файлы создаются, хра-

» **Месяц назад** Вращая слои на холсте, мы запросто создали геометрический дизайн.

фотоаппаратом

Мультимедиа-карты весьма многообразны, но большинство фотокамер имеют гнездо для обычного кабеля USB.

нятся и передаются по физическому соединению. Обычно PTP работает с USB-соединением, что определило его альтернативное название USB PTP. Когда цифровые камеры появились на рынке, PTP был первым методом передачи файлов с камеры на компьютер. Это означает, что если у вас старая цифровая камера, то вам следует использовать инструменты, поддерживающие PTP-доступ к вашим фотографиям.

PTP поддерживается в Linux при помощи библиотеки *libgphoto2*, лежащей в основе инструментов вроде *GPhoto* или *DigiKam*. Эти программы предоставляют графический интерфейс с перетаскиванием мышью, предпросмотр фотографий и все удобства современных фото-менеджеров. Хорошая новость: библиотека *libgphoto* поддерживает массу старых камер, а стало быть, весьма вероятно, что особых проблем с вашими фотографиями не будет.

Во многих камерах все еще используется PTP, хотя большинство добавляет и второй способ передачи: протокол USB Mass Storage, также известный как USB MSC. Он позволяет устройству – вашей камере – выглядеть как еще один диск компьютера. Так что передача снимков – обычное копирование файлов с одного «диска» на другой, все равно как если бы вы вставили карту памяти вашего фотоаппарата в компьютер; только внешним жестким диском будет выглядеть сама камера.

PTP используется в камере или USB MSC, для среднего пользователя безразлично. Главное – выудить свои снимки. Давайте начнем с камер, имеющих USB-гнездо.

Извлечение фотографий с камеры

Все камеры семейства Хэммелов подключаются к компьютеру при помощи USB-кабелей. Они обнаруживаются на USB-шине, но не распознаются как USB MSC при включении. Для работы с ними я мог бы настроить hotplug, но это уж последнее средство. И я решил попробовать *libgphoto2* или использовать мультимедиа-карту. Я начал с *DigiKam* – приложения, основанного на KDE и *libgphoto2*.

Проще быть не может. Подключите USB-кабель к камере и компьютеру, затем запустите *DigiKam*. Включите камеру и настройте ее на режим Показа изображений [Image Display]. В *DigiKam* перейдите в Камера > Просмотреть носитель > USB Imaging Interface. После некоторой задержки, когда *DigiKam* подключается к камере и загружает миниатюры, откроется окно предпросмотра. Теперь мы готовы к работе с инструментами управления фотографиями *DigiKam*.

При наличии подходящего драйвера я могу просто вставить карту в кард-ридер, и она автоматически отобразится в **/dev/mmcblk0**. Чтобы смонтировать это устройство, нам потребуется доступ к его первому разделу, т.е. к **/dev/mmcblk0p1**.

```
mkdir /mnt/media
mount /dev/mmcblk0p1 /mnt/media
```

Так я смогу получить доступ к фотографиям напрямую с флэш-носителя. Теперь, раз вы добрались до файлов вашей камеры, не худо кое-что о них узнать. Первое и главное: форматы. Практически все цифровые камеры поддерживают JPEG. Этот формат удобен для тех, кто не занимается серьезной обработкой фотографий, но он использует кодирование с потерями и нередко (по крайней мере, частично) сжимается программным обеспечением камеры, что не идеально для процессов редактирования изображений.

DigiKam **можно сообщить тип искомой фотокамеры, хотя моей Canon PowerShot этого не потребовалось.**

Советы по медиа

- Работа с мультимедиа-картами в ядрах до 2.6.23 – нечто вроде рулетки. Однако ядра 2.6.24 и новее предоставляют гораздо более качественную поддержку, и многие кард-ридеры работают прямо из коробки.
- Кард-ридеры, основанные на TI (использующие драйвера *ti_xxx*), имеют некоторые проблемы с ядрами до 2.6.24. Похоже, что в ядрах 2.6.24 и новее это исправлено.
- Поскольку поддержка данных носителей не так стабильна, как устройств других типов, вы можете ограничиться чтением с карт; удалять изображения с них лучше в камере.
- Флэш-носители имеют ограниченное число циклов записи, так что перед обработкой фотографий лучше скопировать файлы на локальный жесткий диск.
- FAQ по *DigiKam* содержит инструкции для доступа к камерам через USB MSC: www.digikam.org/drupal/node/25.

Некоторые камеры поддерживают TIFF – это формат без потерь, и он допускает сжатие. В случае формата с потерями при сжатии теряется часть информации об изображении, хотя визуально картинка выглядит как оригинальная, несжатая версия. Формат без потерь не теряет никаких данных. Поэтому распаковка сжатого изображения TIFF даст оригинал, чего не произойдет при распаковке сжатого JPEG-изображения.

Но ни один из этих форматов даже близко не дотягивает до качества изображения, которое вы можете получить из родного формата файлов камеры. Общее название для него – RAW [англ. сырой], и он свой у каждого изготовителя. Как правило, формат RAW используется только в памяти фотоаппарата, но не на карте памяти.

RAW-снимки больше подходят для обработки в *Gimp* или другом графическом ПО, т.к. при сохранении в них не было никакой потери данных (при конвертировании RAW в TIFF информация может быть частично утрачена, несмотря на сжатие без потерь). Поскольку у большинства камер возможности обработки слабые, для преобразования RAW в TIFF или JPEG они используют менее сложное ПО, чем инструменты, имеющиеся на настольном компьютере. Так что работа с RAW, вообще говоря, считается единственным вариантом для профессиональных фотографов.

При этом заполучить RAW-изображения не всегда просто. Для *Gimp* есть модуль расширения (*UFRaw*), способный их обработать, но вообще-то он не извлекает снимки с камеры. И не все камеры сохраняют RAW-файлы на медиа-карте. Вдобавок RAW-файлы больше, чем их JPEG-аналоги, и обычно требуют для работы больше памяти и лучший процессор. Поэтому среднему потребителю, вероятно, больше подойдут JPEG-изображения, особенно если степень сжатия фотографий в вашей камере можно настраивать. И вновь, JPEG отнюдь не идеален, но в бюджетных моделях потребительских камер может быть единственным вариантом.

Упростите себе жизнь

Вы поймете, что искать изображения намного проще, если вы импортируете их в *F-Spot*, используя специфичные тэги. После импорта под каждым изображением будет иконка, представляющая собой связанные с ним метки. Импортированные файлы можно открывать прямо в *Gimp* или других программах вроде *Showfoto* и *KuickShow* в KDE. F-Spot поддерживает различные функции редактирования, которые большинство людей сочтет полезными, включая удаление красных глаз и выравнивание изображения.

ПО для фотокамеры

Имеется три уровня ПО, используемого для цифровых камер. Первый – это *libgphoto2*, применяемая для доступа к файлам камеры при помощи одного из инструментов второго уровня, вроде *DigiKam*. Среди альтернатив *DigiKam* – *GPhoto2*, интерфейс командной строки для *libgphoto2* (для тех, кто точно знает, что он ищет в камере), и *GTkam*, графическая оболочка для *libgphoto2*, работающая так же, как *DigiKam* (хотя с более скудным набором функций). Однако *DigiKam* (и *F-Spot*) – лучшие инструменты управления фотографиями с одинаковой внутренней поддержкой *libgphoto2*, и *GTkam* часто отсутствует на установочных носителях дистрибутивов Linux. Даже если он есть, большинство пользователей сочтут *DigiKam* или *F-Spot* проще в работе после того, как изображения перенесены на компьютер.

Главная альтернатива *DigiKam* – *F-Spot*, приложение для Gnome. Как и *DigiKam*, *F-Spot* можно использовать для извлечения файлов из камеры или для работы с коллекцией изображений на жестком диске. Импорт с моей Canon PowerShot был столь же прост, как и в *DigiKam*. Сначала я включил камеру, а затем выбрал Файл > Импорт и в открывшемся диалоге указал камеру в качестве источника. После этого *F-Spot* подключился к фотоаппарату и начал создавать список миниатюр фотографий для импорта.

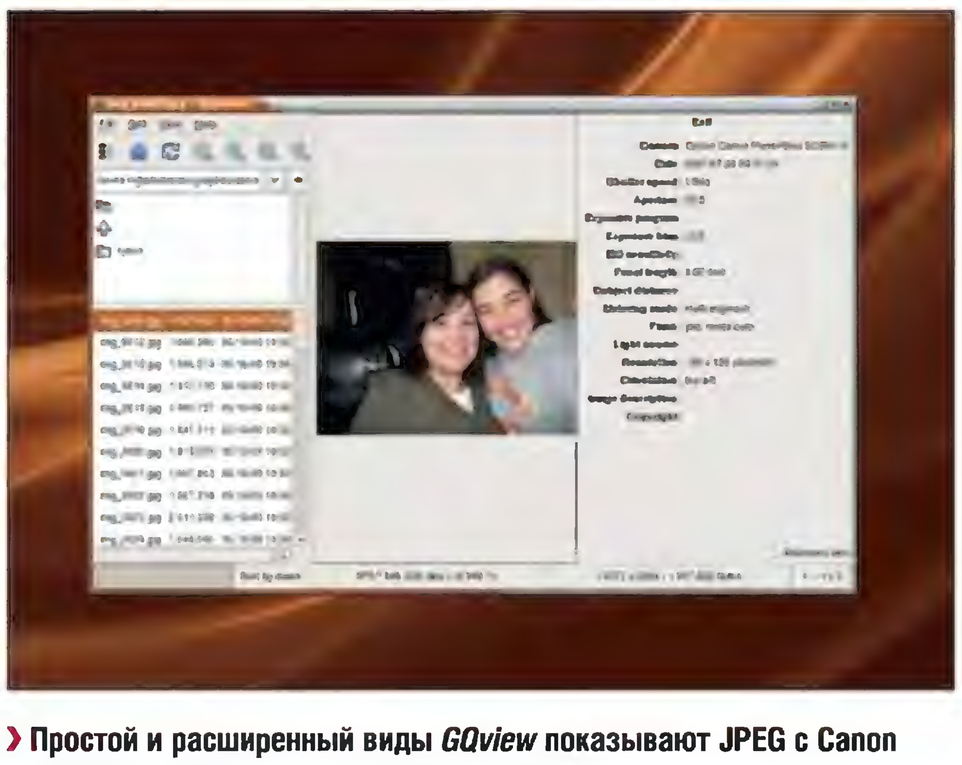
Для разбиения снимков на категории *F-Spot* применяет тэги. Вы можете присвоить всем импортируемым файлам один и тот же тэг, созданный до начала импорта. Позднее каждую фотографию можно снабдить добавочными метками или изменить существующие. Присвоение тэгов позволяет искать изображения на основе текстового описания (тэгов).

F-Spot обладает интегрированной поддержкой создания фото-CD из вашей коллекции изображений. Это упрощает распространение ваших снимков без необходимости запуска другого инструмента для прожига диска – функция, которой пока нет в текущей версии *Gimp*.

Среди других возможностей *F-Spot* – экспорт в web-галереи изображений, вроде Picasa Web и Flickr. Имеется также встроенная поддержка слайд-шоу и полноэкранного режима, и *F-Spot* умеет отправлять изображения по почте, используя настроенную в Gnome программу, например, *Evolution* или *Thunderbird*.

Большинство камер записывают некоторую текстовую информацию о фотографии в файл изображения. Обычно это реализуется в виде метаданных формата EXIF (Exchangeable Image File). Альтернативные форматы метаданных – XMP (Extensible Metadata Platform) и IPTC (International Press Telecommunications Council); ведущие изготовители камер движутся в направлении XMP.

› Интерактивное выравнивание и сепия – только два из небольшого набора инструментов редактирования изображений, встроенных в *F-Spot*. Чтобы вернуться к оригиналу, выберите Файл > Версии > Оригинал [File->Version->Original].



› **Простой и расширенный виды *GQview* показывают JPEG с Canon PowerShot, а заодно и массу метаданных.**

Доступ к метаданным фотографии в формате EXIF не включен в сонм утилит *Gimp* или модулей расширения. Для просмотра данных EXIF можно воспользоваться инструментом вроде GQview или утилитой командной строки *exiftool*. EXIF-данные по умолчанию сохраняются вместе с файлами JPEG, хотя разрешение может быть изменено в соответствии со стандартными настройками для него, установленными в *Gimp*.

Обработка фотографий в GIMP

Gimp может использоваться для редактирования фотографий, хотя и *DigiKam*, и *F-Spot* справляются со следующими операциями так же легко. Большинство пользователей удивятся, когда вы станете использовать *Gimp* вместо этих инструментов. Отвечайте, что вы хотите скорректировать цвета изображения вручную, совместить кадры или выполнить сложные операции вроде смены глубины резкости или цвета. Однако эти операции слишком сложны, чтобы описать их в этом месяце. Рассмотрим базовые действия, которые можно выполнить в *Gimp*.

Улучшение цвета – стандартное требование цифровых фотографий. Проблема здесь в том, что в момент съемки предмет был плохо освещен. Инструменты автоматической коррекции цвета есть и в *DigiKam*, и в *F-Spot*, но *Gimp* справляется с этим лучше.

На данной фотографии одной из моих собак цвет как-то размыт. Пропустив снимок через коррекцию цвета в *F-Spot*, получаем изображение лучшего качества, но *Gimp* дает более точные цвета. В обоих случаях подгонялись уровни и цветность, но в *Gimp* дополнительный этап микширования каналов дает лучший результат.

Выравнивание изображения требуется, когда предмет фотографии находится под углом (возможно, мнимым) к линии горизонта. Для этого необходимо вращение, но вращение прямоугольного изображения выливается в большее изображение с областями, заполненными цветом фона (или прозрачными). Поэтому выравнивание выполняется в два шага: вращение, а затем кадрирование.

Вращение в *Gimp* с версии 2.4 стало намного проще, поскольку справа в окне изображения имеется предпросмотр. На этом изображении перила позади собаки наклонены. Мы можем выпрямить их, перенеся направляющую к верхней части перил у правой стороны изображения, а затем, используя инструмент Вращение, сделав границу параллельной направляющей.

После вращения изображение будет содержать по углам прозрачные участки, и они должны быть обрезаны. Чтобы сберечь все пиксели оригинала изображения после вращения, оно увеличивается, и там, где необходимо, добавляется прозрачность.

› ***F-Spot*** **хорошо корректирует цвета на размытом изображении, но лучший результат был достигнут благодаря изобилию соответствующих инструментов в *Gimp*.**

Удаление красных глаз

В *Gimp* 2.4 чрезвычайно просто исправить красные глаза. На этом снимке вспышка камеры отразилась от задней стенки глаза, вызвав появление красных зрачков. Выделение глаз выполняется при помощи инструмента Свободное выделение [Free Select] (осторожно – не прихватите кожу, она также бывает красного оттенка). Откройте фильтр Удалить эффект красных глаз (Фильтры > Улучшение > Удалить эффект красных глаз). Стандартные настройки сработали для данного снимка хорошо, так что дополнительных изменений не понадобилось.

Как видите, *Gimp* превосходен в редактировании изображений, тогда как для управления фотоколлекцией необходимы другие инструменты. К счастью, мир Open Source полон программами, пригодными для этой работы. Все, что нужно теперь – это ваш талант созидателя.

LXF

› **Красные глаза вызываются отражением света от задней стенки глаза – обычная проблема неопытных фотографов; но GIMP легко это исправит.**

» **Через месяц** Мы склеим превосходную картинку из множества стандартных изображений.

Rails: Добавим

Часть 2 Простейшие возможности галереи реализованы; **Алекс Янг** приобщит вас к Web 2.0 через супер-технологии Ajax и вспомогательные классы Rails.

На данном этапе наша галерея на Rails поддерживает регистрацию пользователей и загрузку фотографий (считая, что вы прошли урок прошлого номера), но в ней кое-чего не хватает – нет нумерации страниц, нет модульного тестирования и нет Ajax, столь любимого Web 2.0-сайтами типа Flickr. Чтобы восполнить пробелы, я расскажу вам о тестировании моделей Rails, использовании вспомогательных классов Rails и крутых технологиях JavaScript, делающих интерфейс более дружелюбным.

Данный урок основан на коде из предыдущей статьи, найти который можно в PDF-файле в разделе **Журнал/Rails** нашего DVD. Класс редактирования на месте средствами JavaScript может пригодиться вам и в других проектах.

› Цель: нумерация страниц, удобочитаемые даты и диалоги редактирования текста, вызываемые щелчком по заголовку.

Наш эксперт

Алекс Янг – технический писатель и ведущий инженер Helicoid Limited, компании, специализирующейся на разработке web-приложений на Ruby on Rails. Алекс регулярно пишет статьи о Rails для сайта **http://alexyoung.org**.

Часть 1 Тесты

Тестирование – неотъемлемая часть процесса разработки на Ruby on Rails. Написание тестов во время изучения Rails также помогает исследовать эту среду, экспериментировать с различными подходами и в конечном счете получить лучший код. Создатели Rails вложили в пакет все необходимое для обучения тестированию, включая несколько задач Rake. До запуска тестов выполните из каталога проекта следующую команду:

```
rake db:test:prepare
```

Она подготовит базу данных *SQLite* в каталоге **db/test.sqlite3** и скопирует схему для нашего проекта. Если вы измените БД посредством миграции, не забудьте выполнить команду **rake db:test:clone_structure** до прогона тестов.

Чтобы запустить все тесты, наберите:

```
rake test:units
```

Когда вы создаете модели и контроллеры с помощью скрипта **generate**, Rails автоматически генерирует заглушки для тестов в каталоге **test/**. Модульные тесты предназначены для моделей, а функциональные – для контроллеров. Первые позволяют проверить, что модели работают как ожидается. Это невероятно удобно, если проект меняется с течением времени: при модификации модели можно убедиться, что сопутствующая система не затронута.

Для тестовых данных Rails использует наработки [fixtures]. По умолчанию они пишутся на YAML и хранятся в каталоге **test/fixtures/**. Наработки могут даже содержать фрагменты на Ruby, как HTML-шаблоны Rails:

```
quentin:
id: 1
login: quentin
email: quentin@example.com
salt: 7e3041ebc2fc05a40c60028e2c4901a81035d3cd
crypted_password: 00742970dc9e6319f8019fd54864d3ea740f04b
1 # test
```

Месяц назад Мы построили фотогалерею с авторизацией пользователя, подгрузкой и правкой фото.

функции Web 2.0

```
created_at: <%= 5.days.ago.to_s :db %>
```

Пример взят из **test/fixtures/users.yml** – обратите внимание, что мы добавили дату при помощи Ruby, с тэгами **<%= %>**.

Покамест фотографии добавляются без заголовков – а какая же галерея без них? Чтобы решить проблему, добавим новую валидацию в **Photo** и протестируем ее. Откройте файл **app/models/photo.rb** и впишите в него строку:

```
validates_presence_of :title, :if => Proc.new { |photo| photo.thumbnail.
nil? }
```

Валидация включает блок проверки, не является ли фото миниатюрой – потому что они спокойно сохраняются и без заголовков: так уж устроен модуль *attachment_fu*. Давайте проверим, что все работает как надо: откройте файл **test/unit/photo_test.rb**. Мы добавим туда несколько новых тестов и удалим заглушку, сгенерированную Rails:

```
require File.dirname(__FILE__) + '/../test_helper'
class PhotoTest < ActiveSupport::TestCase
  fixtures :photos
    def test_titles_are_required_on_create
      assert Photo.create.errors.on(:title)
    end
    def test_titles_are_required_on_update
      photo = photos(:one)
      photo.title = nil
      photo.save
      assert photo.errors.on(:title)
    end
    def test_titles_are_not_required_for_thumbnails
      assert photos(:thumbnail).valid?
    end
```

Обратите внимание, что все тестовые методы имеют префикс **test_** и включают один или несколько вызовов **assert**. Все методы, имя которых начинается с **test_**, запускаются тестовым пакетом автоматически.

Теперь нам недостает лишь распознавания миниатюр. Откройте файл **test/fixtures/photos.yml** и дополните его следующим кодом YAML:

```
thumbnail:
description: MyText
filename: MyString
content_type: MyString
size: 1
user_id: 1
width: 1
height: 1
thumbnail: example.png
content_type: image/png
```

Чтобы тест завершился успешно, все утверждения [assertions] должны быть истинными. Многие из них предоставляются как Rails, так и библиотекой Ruby **Test::Unit**. Мы взяли простейшее: оно проверяет, что выражение не возвращает nil или false. Строка с fixtures использует файл **test/fixtures/photos.yml** для создания фотографий в базе данных. Доступ к «наработке» обеспечивается методом **photos()** – он любезно сгенерирован для нас. В примере выше используется **photos(:one)**. Тесты фото вызываются командной строкой **ruby test/unit/photo_test.rb**, а можно запустить все модульные тесты сразу, набрав команду **rake:test:units**.

Часть 2 Ajax и вспомогательные классы Rails

Нашим приложением было бы проще пользоваться, имей оно пару функций, реализованных на JavaScript. Rails предоставляет богатый функционал для работы с JavaScript через каркас *Prototype* и библиотеку *Scriptaculous*. Обработка текста и HTML реализуется библиотеками Ruby *ActiveSupport* и *ActionView*, поставляемыми с Rails. Воспользуемся вспомогательными классами Rails для создания удобных дат, а также заголовков и описаний с редактированием на месте, как во Flickr. Выводом при просмотре тоже будут управлять вспомогательные классы. На прошлом уроке вы могли заметить вызовы **link_to, image_tag** и **form_for**. Эти методы предоставляются различными вспомогательными классами внутри **ActionView::Helpers**. Прежде чем самим писать вспомогательные функции, всегда полезно проверить, нет ли их в **ActionView**, так как в Rails их сотни.

В нашей галерее фотографии отображаются без всяких дат, поэтому откройте файл, касающийся фото (**app/views/photos/_photo.html.erb**), и добавьте следующую строку после тэга заголовка **<h2>**:

```
<h3>Added on <%= photo.created_at %></h3>
```

К сожалению, это выглядит не очень хорошо. Дата содержит информацию о времени и часовом поясе, а хватило бы только числа, месяца и года. Измените эту строку на такую:

```
<h3>Added on <%= photo.created_at.to_date.to_s :long %></h3>
```

Теперь дата отображается в британском полном формате: Месяц День, Год. Это работает благодаря методу Rails **to_formatted_s**, с синонимом **to_s**. И даты, и времена работают подобным образом, поэтому значение, сохраняемое в **created_at**, преобразуется в дату, а время размещения фото игнорируется.

Чтобы узнать больше о форматах даты и о том, как добавить свой собственный, выполните следующую команду для вывода документации Ruby по теме:

```
ri ActiveSupport::CoreExtensions::Date::Conversions
```

Скорая помощь

Использование блоков **respond_to** в контроллерах позволяет Rails разумно отвечать на запросы различных форматов. Они лежат в основе XML Rest API и респондеров Ajax.

Живые заголовки

Хотя функция «Редактировать на месте» выглядит весьма продвинутой, реализовать ее довольно просто. Большую часть работы сделает библиотека *Scriptaculous*, по умолчанию поставляемая с Rails, но **PhotosController** нужно немного адаптировать, чтобы он смог работать с тем, чего ожидает *Scriptaculous*. Во-первых, нужно приспособить действие **PhotosController#update** для работы с JavaScript. Отредактируйте действие обновления в файле **app/controllers/photos_controller.rb**:

```
def update
  @photo = current_user.photo
  respond_to do |format|
  if @photo.update_attributes params[:photo]
    format.html do
```

Вспомогательные классы Rails: Быстрая справка

textilize text	Конвертирует текст в HTML с помощью *Textile* (установите gem-пакет *RedCloth*, чтобы получить эту функциональность).
truncate text, length	Если текст длиннее, чем length, он будет обрезан и дополнен '...'.
tag name	Создает HTML-тэг, самозакрывающийся, если это необходимо.
content_tag name, content	Создает блок с тэгом HTML, например:**content_tag :p, "Example"**
number_to_human_size size	Преобразует размер файла в байтах в читаемый формат, при необходимости добавляя KB, MB, GB.

```
    flash[:notice] = 'Photo was successfully updated.'
    redirect_to(@photo)
   end
   format.xml { head :ok }
   format.js do
    if params[:field]
     render :inline => '<%= h photo.attributes[params[:field]]} %>'
    else
     head :ok
    end
   end
  else
   format.html { render :action => "edit" }
   format.xml { render :xml => @photo.errors, :status => :
unprocessable_entity }
   format.js  { render :text => @photo.errors.full_messages, :status
=> :unprocessable_entity }
  end
 end
end
```

Этот код делает кое-что новое. Обработка запросов JavaScript поддерживается в **format.js** в блоке **respond_to**. Когда фотография успешно сохранена и используется JavaScript, выводится заголовок, поскольку **Ajax.InPlaceEditor**, предоставляемый библиотекой *Scriptaculous*, ожидает, что сервер возвратит обновленный текст. Обратите внимание, что для этого используется **render :inline**, позволяя контроллеру вызвать вспомогательный метод. Текст экранируется, чтобы избежать опасных метасимволов. Также добавьте в начало контроллера (после строки **before_filter**) следующую строку:

```
skip_before_filter :verify_authenticity_token, :only => [ :update ]
```

По соображениям безопасности в Rails используется механизм аутентификации, порождающий проблемы при обновлении с помощью JavaScript, и пока мы его отключим.

Позаботимся о редактировании

Нам надо, чтобы редактирование на месте предоставлялось только зарегистрированным пользователям, но чтобы проверить это, не будем забивать наш файл выражениями с **if**, а создадим новый вспомогательный класс. Откройте файл **app/helpers/application_helper.rb** и добавьте следующий метод:

```
def in_place_editor_if(condition, object, field, &block)
  if condition
    object_id = "#{object.class.name.downcase}_#{field}_#{object.
id}"
    in_place_class = "in_place_#{field}"
    concat '<span id="%s" class="%s">' % [object_id, in_place_
class], block.binding
    yield
concat '</span>', block.binding
  else
    yield
  end
end
```

Это довольно-таки продвинутый Ruby – ничего страшного, если понять его сейчас трудно, но обязательно вернитесь к нему позже! Его нужно использовать примерно так:

```
<% in_place_editor_if current_user, photo, :title do %><%= h
photo.title %><% end %>
```

Если переменная **current_user** установлена, заголовок фото будет обернут в тэг **** с атрибутами id и class, для упрощения ссылок на него из JavaScript. У незарегистрированных пользователей заголовок будет выводиться без тэга ****, а значит, JavaScript его проигнорирует. В данном фрагменте кода заголовок упрятан в блок с синтаксисом скобок **do … end**. Таким образом, этот вспомогательный класс может инкапсулировать все, даже многострочный HTML. Внутри вспомогательного класса для доступа к этому HTML-коду и добавления в него дополнительного текста вызываются **yield** и **concat**.

Хотя это уже высший пилотаж, настоятельно рекомендую вам применять его и в других вспомогательных методах, которые вы соберетесь писать. Отредактируйте файл **app/views/photos/_photo.html.erb**, чтобы употребить в нем этот хитрый вспомогательный класс для изменения способа отображения заголовков:

```
<h2>
<% in_place_editor_if current_user, photo, :title do %><%= h
photo.title %><% end %>
<%= link_to 'Edit', edit_photo_path(photo) %>
<%= link_to 'Delete', photo, :confirm => 'Are you sure?', :method
=> :delete %>
</h2>
```

Это упростит поиск HTML в JavaScript и применит редакторы к каждому заголовку.

Добавим редактирование через JavaScript

Последний кусочек мозаики – код JavaScript для редактирования на месте. Прежде чем писать его, поправим HTML-код основного файла и подключим туда наши скрипты. В Rails для этого есть вспомогательный класс **javascript_include_tag** – добавьте его в секцию **<head>** файла **app/views/layouts/application.html.erb**:

```
<%= javascript_include_tag :all %>
```

Добавьте следующий код в файл **public/javascripts/application.js**:

```
var EditorGenerator = Class.create({
 initialize: function(class_name, url_base, error_message) {
  this.item_selector = '.' + class_name
  this.update_url = '/' + url_base + '/update/'
  this.error_message = error_message
  this.field = class_name.replace(/in_place_/, '')
  this.create_editors()
 },
  create_editors: function() {
   $$(this.item_selector).each(function(element) {
    var item_id = element.id.match(/_(\d+)$/)[1]
    this.create_editor(element, item_id)
   }.bind(this))
  },
 create_editor: function(element, item_id) {
  new Ajax.InPlaceEditor(element, this.update_url + item_id +
'?field=' + this.field, {
   onFailure: this.onFailure.bind(this),
  callback: function(form, value) { return 'photo[' + this.field + ']='
  + escape(value) }.bind(this)
   })
 },

  onFailure: function(editor, transport) {
  alert(this.error_message + ': ' + transport.responseText)
```

```
}
})
document.observe('dom:loaded', function() {
 var title_generator = new EditorGenerator('in_place_title',
'photos', 'Error saving photo')
})
```

Этот код создает класс JavaScript, применяющий редактор **Ajax.InPlaceEditor** к заданным HTML-элементам. Функционал, который здесь используется, предоставляется библиотеками JavaScript *Scriptaculous* и *Prototype* – они у нас есть. Класс инициализируется вызовом document.**observe('dom:loaded', ...)**, который ждет готовности браузера, прежде чем выполнить какой-либо JavaScript. Как только класс **EditorGenerator** будет создан, он добавит **Ajax.InPlaceEditor** в каждый из span'ов, которые сгенерировал наш вспомогательный класс. Другая интересная строка кода – это итератор, используемый в цикле для прохода по каждому заголовку:

```
$$(this.item_selector).each(function(element) { ...
```

В нем используется функция *Prototype* **$$()** для выбора списка элементов на основе селектора CSS. Он хранится в свойстве **this.item_selector**, устанавливаемом при инициализации класса. Функция **$$()** возвращает массив элементов, и, значит, для прохода по нему можно использовать метод **each()**, также предоставляемый *Prototype*. Это итератор, и он считается лучше читаемым по сравнению с обычным циклом.

Теперь можно добавить редактируемые поля всюду, где отображаются заголовки. В качестве примера откроем файл **app/views/photos/show.html.erb** и изменим его заголовок на этот:

```
<h2><% in_place_editor_if current_user, @photo, :title do %><%=
h @photo.title %><% end %></h2>
```

Другой хороший кандидат на редактирование – описания. Измените их в **app/views/photos/show.html.erb** таким образом:

```
<p><% in_place_editor_if current_user, @photo, :description do
%><%= textilize_without_paragraph h(@photo.description) %><%
end %></p>
```

Потом замените описание в файле **app/views/photos/_photo.rhtml** на следующее:

```
<p><% in_place_editor_if current_user, photo, :description do
%><%= textilize_without_paragraph h(photo.description) %><%
end %></p>
```

Затем добавьте еще один объект **EditorGenerator** в **public/javascripts/application.js**, после **title_generator**:

```
var description_generator = new EditorGenerator('in_place_
description', 'photos', 'Error saving photo')
```

То же самое можно сделать с любым другим полем. Есть лишь небольшая загвоздка: если кликнуть по ссылке для редактирования описания, текст будет представлен в виде HTML-кода. Это можно поправить, добавив в **Ajax.InPlaceEditor** параметр **loadingText**, указывающий на действие в **PhotosController**, которое удалит ненужную разметку.

Скорая помощь

Для отладки JavaScript или даже исследования объектной модели документа DOM или CSS страницы, настоятельно рекомендую модуль расширения *Firefox* под названием *Firebug* (**https://addons.mozilla.org/en-US/firefox/addon/1843**). Без него причины ошибок в JavaScript будет понять очень трудно.

Часть 3 Разбивка на страницы

Чтобы достойно завершить наш второй урок по разработке галереи, добавим функцию постраничного вывода. Как ни странно, решение этой задачи в web-приложениях сопряжено с трудностями, так как случайно написать увесистый SQL-запрос очень просто. К счастью, Мислав Марохнич [Mislav Marohnic] создал невероятно простой модуль расширения **will_paginage**, удовлетворяющий требованиям большинства приложений.

Предпочтительный метод установки – с помощью gem:

```
sudo gem1.8 install mislav-will_paginate --source http://gems.
github.com/
```

Теперь откройте файл **config/environment.rb** и добавьте в него следующие строки в конце:

```
gem 'mislav-will_paginate', '~> 2.2'
require 'will_paginate'
```

Перезапустите сервер, чтобы приложение загрузило модуль.

Пользоваться им очень просто. Откройте **app/controllers/photos_controller.rb** и измените индексирование при загрузке фотографий таким образом:

```
@photos = Photo.paginate :all, :conditions => 'thumbnail is
NULL', :page => params[:page], :order => 'created_at DESC'
```

Теперь откройте файл **app/views/photos/index.html.erb** и добавьте туда строку:

```
<%= will_paginate @photos %>
```

Все почти готово, но чтобы разбивка на страницы хорошо читалась, нужно немного приятного CSS. Добавьте следующий код в файл **public/stylesheets/screen.css**:

```
.pagination { text-align: center; padding: .3em; clear: both; margintop:
1em; width: 100%; float: left; }
.pagination a, .pagination span { padding: .2em .5em; }
.pagination span.disabled { color: #AAA; }
.pagination span.current { font-weight: bold; color: #FF0084; }
.pagination a { border: 1px solid #DDDDDD; color: #0063DC;
textdecoration: none; }
.pagination a:hover, .pagination a:focus { border-color: #003366;
background: #0063DC; color: white; }
.pagination .page_info { color: #aaa; padding-top: .8em; }
.pagination .prev_page, .pagination .next_page { border-width: 2px; }
.pagination .prev_page { margin-right: 1em; }
.pagination .next_page { margin-left: 1em; }
```

Пока вы не добавите не менее 30 фотографий, номеров страниц вам не видать. Для тестовых целей добавьте опцию **per_page** в действие **index**:

```
@photos = Photo.paginate :all,
:conditions => 'thumbnail is NULL',
:page => params[:page],
:per_page => 5
:order => 'created_at DESC'
```

Итак, мы изучили основы модульных тестов, разбивки на страницы, Ajax и объектно-ориентированный JavaScript; можете применить эти технологии в своих проектах. Библиотеки JavaScript *Scriptaculous* и *Prototype* имеют стабильное API, поэтому у вас будет возможность взять пример редактирования на месте и создать на его основе более сложные элементы управления. Модульное тестирование Rails поможет убедиться в работоспособности вашего кода. Вы почти полностью вооружены для профессиональной разработки на Rails! LXF

Комплект библиотек JavaScript

Rails поставляется с библиотеками *Prototype* и *Scriptaculous*. *Scriptaculous* (**http://script.aculo.us**) построена на *Prototype* (**http://prototypejs.org**). На сайтах обеих библиотек есть подробная документация. *Prototype* упрощает использование JavaScript на различных платформах, а также включает средства объектно-ориентированной разработки, библиотеку Ajax и абстракцию DOM для упрощения навигации в HTML-документах из ваших программ. *Scriptaculous* предоставляет средства для анимации, эффектов, drag and drop, сложные элементы управления Ajax и модульное тестирование.

Python: Создаем

Python и Qt идеальны для быстрой разработки приложений. **Грэм Моррисон** докажет это, используя новейшие технологии KDE и примерно 60 строк кода.

Наш эксперт

Грэм Моррисон
Разработчик свободного ПО, фанат KDE и энтузиаст электронной музыки. Он обсуждал синтезаторы не с одним ведущим членом сообщества FOSS.

В прошлом месяце мы связали Python и синтезатор речи *Festival* для создания читалки RSS, озвучивавшей слова из RSS-ленты. Если вы прошли тот урок полностью, то уже знаете, что *Festival* – синтезатор достойный, но образцом дружелюбия к пользователю не является. Даже простейшие задачи, вроде печати слов и их последующего чтения, требуют множества аргументов и загадочного формата командной строки. Нет, это не выход. Для *Festival* явно нужны простейший графический интерфейс с полем для печати слов и большая кнопка «ЧИТАТЬ». Именно его мы и создадим на данном уроке.

Python и Qt

Чтобы облегчить себе жизнь, воспользуемся тем же средством быстрой разработки, что и в прошлом месяце – Python. Несмотря на свою скриптовую природу, Python отлично подходит и для создания приложений с графическим интерфейсом. В нем не нужно беспокоиться о всякой ерунде типа объектов, типов и инициализации, тормозящей разработку графических интерфейсов в других языках. Программы на Python эффективны и легко модифицируются, и запускать их можно немедленно. Полностью работоспособные приложения могут уложиться в несколько строк кода – эквивалентам на C или C++ потребуется во много раз больше. Однако по умолчанию Python не предусматривает средств для создания графических интерфейсов. Как и в случае RSS-читалки из прошлого месяца, нужная функциональность обеспечивается добавочными модулями; по части GUI их имеется несколько. Совместно с Python обычно используются два кросс-платформенных инструментария – *wxWidgets* и *GTK*, оба могут быть внедрены в ваш собственный Python-скрипт импортированием модулей *WXPython* и *PyGTK* в его начале. Но мы-то возьмем *PyQt*, другой популярный GUI-модуль; как следует из названия, он позволяет применять *Qt* внутри скриптов Python. Тот же инструментарий использован для рабочего стола KDE, и его API битком набит возможностями, от вывода 3D-графики до проигрывания музыки. Месяц назад мы делали обзор последнего релиза *Qt* (4.4), а команда разработчиков *PyQt* уже успела интегрировать эти новшества в последний релиз *PyQt*. Итак, программисты на Python получили преимущества нового *Qt*, включая виджет браузера *WebKit*.

Это очень важно для нашего проекта, потому что мы воспользуемся одной из новых возможностей, чтобы заставить *Festival* заговорить. Имя ее – *Phonon*, это мультимедиа-инструментарий KDE4, здорово облегчающий работу с мультимедиа-потоками. С самим KDE мы иметь дела не будем. *Phonon* портирован обратно в *Qt*, и его функциональность можно применять, обойдясь без разработки полновесного KDE-приложения. До *Phonon* не существовало стандарта кросс-платформенного (или даже кросс-интерфейсного) метода работы с аудио- и видео-потоками. Это раздражало и делало работу программиста гораздо сложнее, чем надо бы. С *Phonon* вам, например, больше не требуется использовать *DirectMedia* в Windows, *QuickTime* в OS X и *GStreamer* в Linux: он выступает как посредник для этих технологий. Вы пишете код для проигрывания, а *Qt* выполняет за вас остальную работу, конвертируя ваш запрос в мультимедиа-инструментарий выбранной вами платформы. Та же философия принята и внутри Python, и поэтому мы выбрали *PyQt* для нашего проекта.

Приложим руки к ушам

Phonon так важен для нашего графического интерфейса синтезатора речи, потому что нам нужно контролировать вывод звука самим. *Festival* включает собственную возможность проигрывания, но она намертво завязана с древней *Open Sound System* (*OSS*). А значит, нельзя поделить аудиокарту с другими приложениями, что приводит к сбоям работы во многих современных дистрибутивах Linux. Чтобы обойти это ограничение, будем записывать вывод от *Festival* в звуковой файл, благо что это не так трудно сделать. Наше приложение будет затем воспроизводить его, для чего мы задействуем *Phonon*.

У этого решения есть два плюса. Во-первых, оно не имеет ограничений, присущих OSS, и с радостью делит аудиокарту с другими приложениями. Во-вторых, пользователь получает возможность перерыва, останова и перемотки синтезируемой речи. Этого трудно достичь без использования аудиофайла, так как синтез лучше не прерывать на полпути. Вывод звука займет лишь доли секунд, затем аудиофайл может быть загружен и проигран. Но все это будет наша Часть 2.

Месяц назад Мы написали читалку RSS на основе синтезатора речи *Festival*.

GUI для Festival

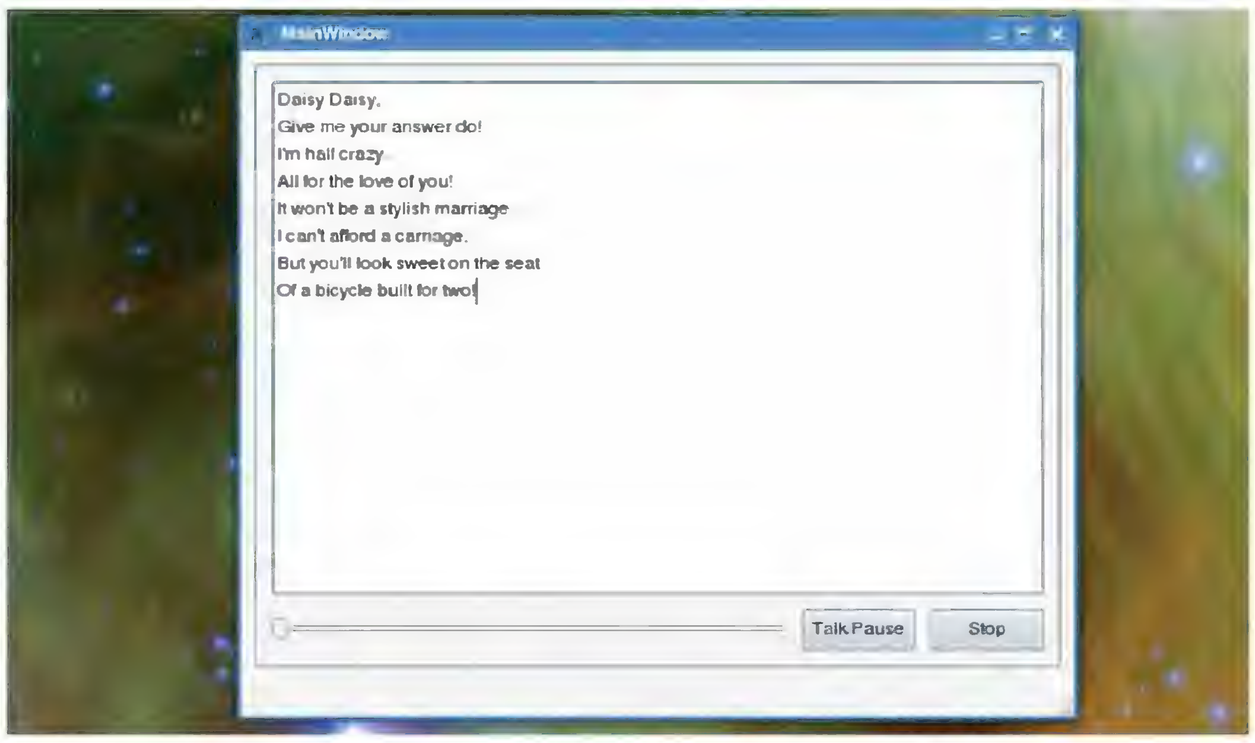
Часть 1 Создаем графический интерфейс

Для данной работы важно предусмотреть следующее. Очевидно, понадобятся установленные и работающие Python и *Festival* – см. урок прошлого месяца, если у вас возникнут сложности с ними. Еще нам необходим SIP (модуль, создающий привязки C и C++ с *Qt*) для Python, также как и *GStreamer*, если они не установлены, а дополнительно – *Qt 4.4* и *PyQt 4.4*. Вы, наверное, уже сможете добавить двоичные пакеты для обоих через менеджер пакетов вашего дистрибутива, но во время написания статьи нам пришлось компилировать и устанавливать и *Qt 4.4*, и *PyQt 4.4* вручную из исходных кодов, что заняло несколько часов. Также вам необходимо убедиться в наличии небольшой утилиты командной строки *pyuic4*; она либо скачивается с исходными кодами *PyQt*, либо имеется в отдельном пакете (**pyqt-dev-tools** для пользователей Debian и Ubuntu).

Итак, приступим к программированию. *PyQt* имеет доступ ко всему API *Qt*, поэтому мы можем использовать *Qt Designer* для разработки пользовательского интерфейса нашего приложения, и создадим в нем основное окно нашей программы. Возможно, куда проще сделать это вручную, но если вы захотите расширить программу, добавив новые возможности и элементы интерфейса, *Designer* сделает этот процесс интуитивнее.

Если вы установили *Qt* из исходных кодов, убедитесь, что переменная окружения **QTDIR** указывает на новое место размещения *Qt 4*. Запустите *Designer* из командной строки или меню «Разработка» на рабочем столе. Проследуйте через все три этапа (*внизу*) для добавления нужных приложению графических элементов, затем сохраните результат в файл в директории проекта. Соглашение требует начинать имя файла с **ui_**, так что определить, где хранится интерфейс пользователя, нетрудно. Файлы *Designer* используют XML для описания схемы только что спроектированного приложения, и теперь нам нужно сконвертировать его в рабочий код Python для построения интерфейса. Инструмент для этого преобразования называется *pyuic4*. Наберите **pyuic4 ui_fooey.ui > ui_fooey.py** для создания кода Python (заменив **fooey** на имя вашего скрипта)..

Обычно *pyuic4* генерирует самостоятельный код, не требующий доработки. Но из-за использования *Phonon*, слишком нового для *Qt Designer*, нужно отредактировать несколько строк в файле **ui_filename.py** вручную. Откройте его в вашем любимом текстовом редакторе и



› Вот так наше приложение должно выглядеть в итоге: большой текстовый виджет с кнопками play/pause и stop. »

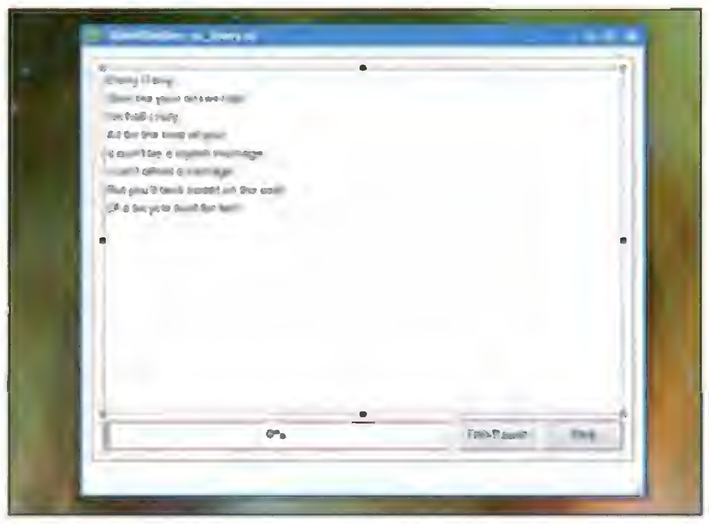
Шаг за шагом: Создаем графический интерфейс

1 Основное окно

Выберите шаблон Main Window в стартовом мастере Designer. Перетащите на него с панели инструментов контейнер-рамку и перенесите в нее два виджета PushButton: TextEdit и ProgressBar.

2 Размещение

Два раза кликните на кнопки для изменения надписей. Мы используем их для воспроизведения/паузы и остановки. Выберите индикатор выполнения и два виджета кнопок, а затем нажмите Ctrl+1 и выровняйте их горизонтально.



3 Компоновка

Можете поиграть с компоновкой виджетов, разместив их так, как вам нравится. Удовлетворившись результатом, нажмите Ctrl+5 для общего выравнивания по внутренней сетке и сохраните файл в директории вашего проекта.

» просто наберите **from PyQt4.phonon import Phonon** в отдельной строке, сразу под первой из строк **from** в начале файла. Так обеспечивается импорт *Phonon* из *PyQt*, и можно использовать его функции совместно с графическим интерфейсом. Поискав ниже, замените три строки, начинающиеся на **self.progressBar**, на следующие две, и поправьте строку **self.horizontalLayout**, чтобы использовать **self.seekSlider**:

```
self.seekSlider = Phonon.SeekSlider(self.frame)
self.seekSlider.setObjectName("progressBar")
self.horizontalLayout.addWidget(self.seekSlider)
```

Здесь мы заменили виджет индикатора выполнения на виджет *Phonon*, что позволит пользователю передвигать и видеть позицию текущего воспроизведения при прослушивании музыкального файла. Если бы *Designer* поддерживал этот виджет напрямую, данный шаг был бы не нужен: мы бы просто перетащили виджет с панели инструментов. Это также означает, что если вы измените компоновку интерфейса (в *Designer*) и снова запустите *pyuic4*, вам придется осуществить все изменения вновь, так как *pyuic4* автоматически перезапишет файл при генерации кода Python.

Вот и весь код, необходимый графическому интерфейсу. Следующий шаг – написание скрипта для использования объектов созданного GUI и добавления «разговорной» функциональности *Festival*.

Часть 2 Логика кода

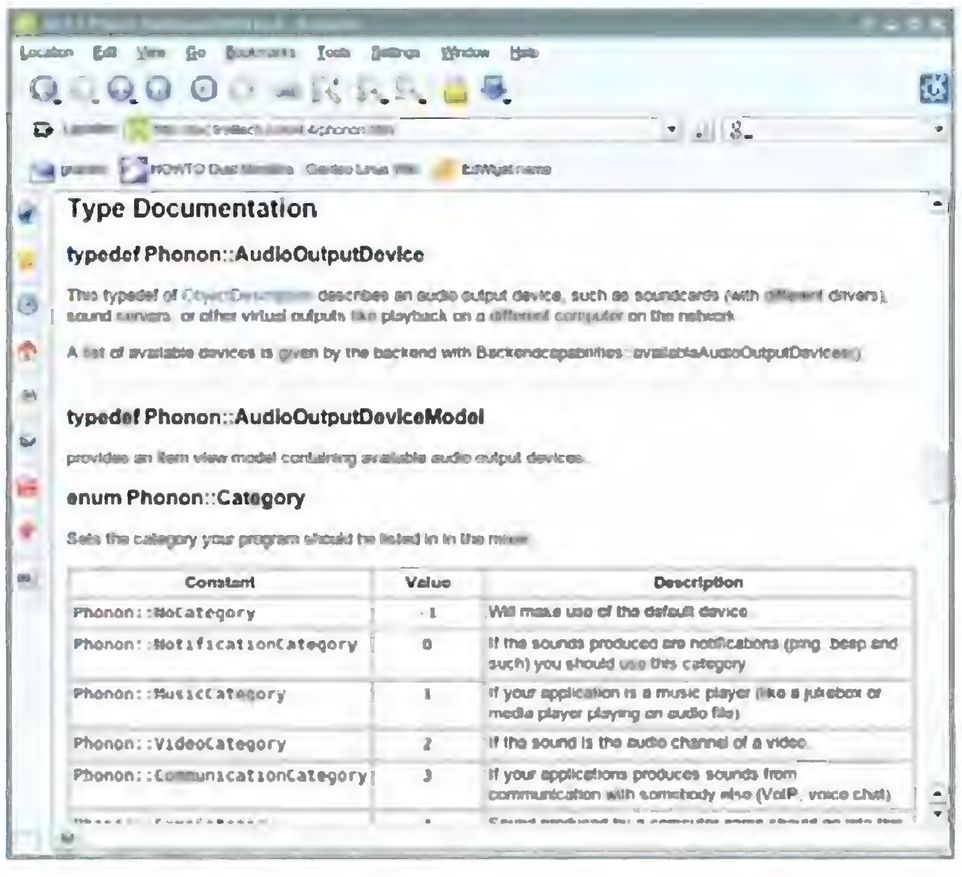
Создайте для остатка нашего скрипта новый текстовый файл. Первым делом опишем модули, планируемые к применению. В начале файла добавьте следующие четыре строки:

```
import sys, os, time, tempfile
from PyQt4 import QtCore, QtGui
from PyQt4.phonon import Phonon
from ui_fooey import Ui_MainWindow
```

Первая добавляет основную функциональность Python. Модуль **sys** мы использовали на прошлом уроке, модуль **os** обеспечивает работу с файлами и путями, **time** понадобится нам позже для добавления небольшой задержки, чтобы мы могли дождаться создания аудио-файла, а при помощи **tempfile** Python автоматически генерирует правильные пути временных файлов в системе. Ниже этой строки мы импортируем основные функции *Qt* из **PyQt4**, а еще ниже – импортируем *Phonon*. Последняя строка импортирует класс GUI, автоматически созданный *pyuic4* из нашего файла *Designer*. Здесь **ui_fooey** – имя нашего файла, а **Ui_MainWindow** – имя класса внутри файла.

Если вы пока не имели дела с именами классов, то их можно представить как обобщение функций, используемое в объектно-ориентированном программировании. Это комбинация данных и функций, их обрабатывающих, сведенная в единую самодостаточную сущность, или «класс». Импорт класса в скрипт Python обеспечит нам доступ к компоновке интерфейса и уже созданным нами виджетам, а также к данным, хранящимся в этих элементах. Мы получим все это, создав новый класс в нашем коде. Под строками импорта добавьте следующее:

```
class StartQT4(QtGui.QMainWindow):
def __init__(self, parent=None):
 QtGui.QWidget.__init__(self, parent)
 self.ui = Ui_MainWindow()
 self.ui.setupUi(self)
```



› Программисты на Python имеют доступ ко всем тем функциям, что и родные разработчики на *Qt*, и без чтения http://doc.trolltech.com/4.4 тут не обойтись.

Сперва мы создаем новый класс с именем **StartQt4** и он описывается как тип **QMainWindow** – его же мы создали и в *Designer*. Это родительский тип окна для всех приложений *Qt*, и это означает нашу возможность управлять окном нашего приложения так же, как и в других программах. Как альтернативу, *Qt* включает виджеты диалогов и окон, но они не часто используются в самостоятельных приложениях. Команда создания следует за описанием функции с именем **__init__** без указания класса.

Как и другие функции Python с похожим названием, она запускается при инициализации объекта и содержит все подпрограммы настройки, которые понадобятся нам для добавления функциональности в ранее созданные виджеты. Далее мы говорим этому новому окну, что интерфейс пользователя управляется из класса основного окна (**self.ui = Ui_MainWindow**), а затем просим *Qt* выполнить компоновку графического интерфейса. Теперь передйем к виджетам в основном окне. Под кодом, приведенным выше, и на таком же уровне отступов добавьте следующие строки:

```
self.audioOutput = Phonon.AudioOutput(Phonon.
MusicCategory, self)
 self.metaInformationResolver = Phonon.MediaObject(self)
 self.mediaObject = Phonon.MediaObject(self)
 self.ui.seekSlider.setMediaObject(self.mediaObject)
 Phonon.createPath(self.mediaObject, self.audioOutput)
```

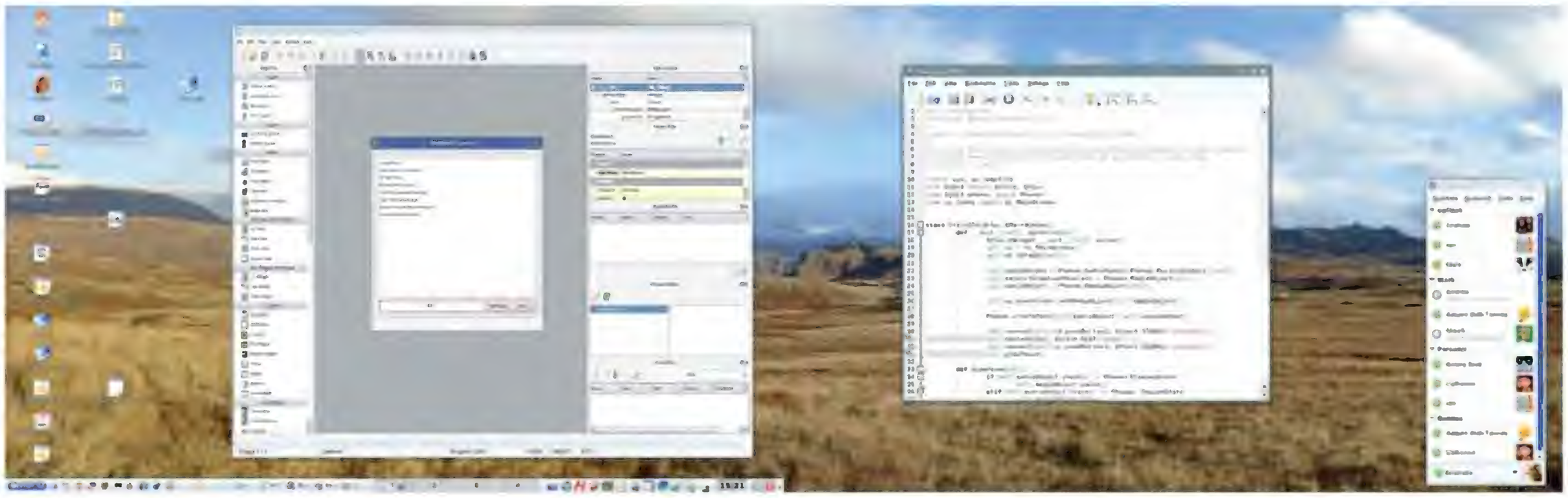
Данный участок кода содержит конфигурацию *Phonon* и настройки нашего приложения. *Phonon* использует несколько соединений для подключения источников, которые мы хотим проиграть, с устройством, на котором мы хотим их проигрывать. Этого достигают первые несколько строк вышеприведенного кода, создавая путь к аудиовыходу через **audioOutput** и метаданные, необходимые *Phonon* для распознавания вида проигрываемого содержимого. Мы связали все это с индикатором проигрывания в графическом интерфейсе и соединили объекты друг с другом. На данном этапе при запуске программы *Phonon* уже находится в стадии готовности к воспроизведению и ждет сигнала к началу работы. Чтобы сделать что-то интересное, добавьте следующие строки:

```
self.connect(self.ui.pushButton2, QtCore.SIGNAL('pressed()'),
self.mediaObject, QtCore.SLOT('stop()'))
self.connect(self.ui.pushButton1, QtCore.SIGNAL('pressed()'),
self.playPause)
```

Тех, кто не привык к *Qt* (как, впрочем, и искушенных программистов), эти строки озадачат: здесь применен механизм слотов и сигналов *Qt*. Сигналы и слоты действуют как указания для управления событиями внутри приложения, прокладывая причинно-следственную связь между различными функциями и объектами программы. Эти связи возникают, когда пользователь нажимает на иконку Сохранить Файл или кликает правой кнопкой мыши на виджет TextEdit, и многие события уже имеют соединения по умолчанию. Так, ползунок Playback автоматически обновляет свое положение в зависимости от места проигрывания аудио-файла, а виджет основного окна Resize автоматически информирует механизм компоновки о необходимости пере-

Скорая помощь

Не вводите весь код за одну сессию. Попробуйте набирать по куску кода, затем запускать для проверки скрипт, чтобы убедиться в его работоспособности. Это упростит отладку.



рисовки дисплея. В верхней строке мы велим приложению вызвать функцию *Phonon* **stop()** (слот), когда кнопка **pushButton2** будет нажата **[pressed()]** (сигнал). Следующая строка вызывает похожую функцию, только в этом случае мы соединяем сигнал кнопки **pushButton1** со слотом – функцией **playPause**. Ее-то мы сейчас и напишем.

```
def playPause(self):
    if self.mediaObject.state() == Phonon.PlayingState:
     self.mediaObject.pause()
   elif self.mediaObject.state() == Phonon.PausedState:
    self.mediaObject.play()
   else:
    os.popen('echo "' + str(self.ui.textEdit.toPlainText()) + '" |
festival_client --ttw > ' + tmpfile, "r")
    time.sleep(0.2)
    self.mediaObject.setCurrentSource(Phonon.MediaSource(str(t
mpfile)))
    self.mediaObject.play()
```

Отступы в **def** нужны такие же, как в **__init__**, где мы начали класс **StartQt4**. Это наша реализация слота **playPause**, и данный код запустится, когда пользователь нажмет на кнопку Play/Pause, поэтому мы сперва проверяем состояние проигрывания внутри кода. Если воспроизведение уже идет, мы ставим его на паузу командой **self.mediaObject.pause**; если же оно приостановлено, мы возобновляем его. При обнаружении состояния, отличного от этих, мы подразумеваем, что проигрывание было остановлено, а слова в текстовом редакторе – изменены. Тогда мы запускаем в командной строке **festival_client**, передаем содержимое текстового поля (**self.ui.textEdit.toPlainText**) и перенаправляем вывод звукового потока во временный файл. Для этого мы используем команду Python *os.popen*. Она берет строку внутри кавычек как аргумент и запускает команду от имени приложения. На время выполнения команды программа останавливается, но нужно подождать несколько секунд для обновления файловой системы, что обеспечивается строкой **time.sleep(0.2)**. Затем мы используем этот WAV-файл как источник для объекта проигрывания *Phonon* и немедленно начинаем воспроизведение.

Осталось сделать только одну вещь: прибрать за собой (э, откуда тут этот временный файл?) и написать функцию main приложения – ту, что запускается первой при его старте.

```
if __name__ == "__main__":
 app = QtGui.QApplication(sys.argv)
 app.setApplicationName("Speech Synth")
 tmpfile = tempfile.mktemp(suffix='.wav')
 os.popen('touch ' + tmpfile)
 myapp = StartQT4()
 myapp.show()
 try:
  sys.exit(app.exec_())
 except SystemExit:
  pass
 os.remove(tmpfile)
```

Первая строка данного кода не должна иметь отступа. Это главная подпрограмма, которая выполнится при запуске нашего скрипта, эквивалент 'int main()' в C и C++. В ней мы генерируем интерфейс из ранее созданного объекта и даем ему имя приложения. Далее мы переходим к созданию временного файла для сохранения «сырых» аудиоданных вывода *Festival*. В Linux это обычно директория **/tmp** в вашем корневом разделе, и мы используем ранее импортированный модуль **tempfile**, чтобы автоматически назначить корректный путь к временной директории для файла, который нам необходимо создать. Мы также снабжаем файл расширением **.wav**, чтобы *Phonon* знал, какой вид данных он проигрывает. Затем мы запускаем в командной строке **touch** для создания пустого файла, который **Phonon** будет использовать как источник для воспроизведения. Наконец, запускаем приложение и показываем графический интерфейс. Приложение теперь зациклено; оно ожидает, когда пользователь закроет его с помощью кнопки в заголовке окна. Когда это произойдет, мы отловим событие в операторе **try** и дадим скрипту Python удалить временный файл перед выходом из программы.

Сохраните этот файл и запустите *Festival* в режиме сервера: **'festival --server'**. Теперь выполните скрипт, набрав **python fooey.py**. Вам будет предоставлен графический интерфейс на *Qt* с полем для ввода текста. Нажатие на Play пошлет текст в *Festival*, а файл результата – назад в *Phonon*, который затем его проиграет. Вы можете ставить на паузу и перематывать файл вперед и назад, используя ползунок. Если у вас кризис идей – что бы такое еще добавить? – гляньте на наш список задач; а кто внесет в программу какие-нибудь грандиозные дополнения, сообщите нам. LXF

› Код *PyQt* гораздо короче, чем эквивалентная программа, написанная на родном для *Qt* C++; на нем легче и тестировать идеи, и создавать быстрые графические интерфейсы.

Список задач

- Кэшировать вывод речи, чтобы одинаковый текст не требовал повторного преобразования, когда пользователь нажимает stop/play.
- Существует проблема безопасности в использовании **tempfile** (он может быть подделан). Решение – использовать **tempfile.mktemp**
- Перехватывать вывод с *Festival* напрямую и использовать его как исходные данные для проигрывания.
- Добавить в интерфейс чтение и запись файлов.
- Обеспечить больший контроль над *Festival*, включая возможность изменять голос диктора.
- Использовать Phonon для добавления эффектов в аудиовывод, например, реверберации, задержки и сдвига тона.

Через месяц Одно из самых сложных приложений верстки: *LaTeX*.

Сияющая

ЧАСТЬ 2 Пожалуй, ни одна из технологий KDE 4 не привлекла к себе столько внимания, как эта. **Андрей Боровский** докажет, что написать собственный плазмоид совсем не сложно.

Если вы уже работали в графической среде KDE 4, то просто не могли не заметить плазмоиды. Для тех, кто все же не знает, что это такое, поясним. Плазмоиды представляют собой улучшенный и украшенный вариант апплетов, известных пользователям KDE с незапамятных времен. Свое название плазмоиды получили от системы Plasma, которая призвана существенно расширить возможности рабочего стола KDE и улучшить его внешний вид. Функциональность Plasma, в свою очередь, основана на новых возможностях библиотеки *Qt 4*, прежде всего – на системе Graphics View Framework (**LXF105**). Помимо нее, Plasma использует новые средства для работы с OpenGL, появившиеся в *Qt 4*. С точки зрения программиста, Plasma представляет собой API, основная часть которого реализована в библиотеке *libplasma*.

Все это будет подробно рассмотрено ниже. Сейчас же мы попробуем ответить на вопрос: стоит ли вам, как разработчику, изучать плазмоиды и заниматься их написанием. Излишне пояснять, что далеко не каждая программа может (или должна) быть выполнена в виде плазмоида. На сегодняшний день их написано немало (список приложений данного типа, известных команде KDE, можно найти по адресу techbase.kde.org/Projects/Plasma/Plasmoids), так что некоторые тенденции уже очевидны. Поскольку плазмоиды «живут» на рабочем столе или на панели задач, основная их часть предназначена для того, чтобы информировать пользователя о различных событиях в системе или за ее пределами (к этой категории относятся часы, измерители сетевого трафика, RSS-читалки и т.п.). Меньшая по численности группа плазмоидов предназначена для решения несложных, но часто востребованных задач (калькуляторы, словари, быстрый запуск программ). Встречаются, впрочем, и плазмоиды-игры, плазмоиды для просмотра комиксов и другая экзотика. С одной стороны – область применения плазмоидов ограничена. С другой, их написание – та сфера, в которой начинающий программист, одержимый потоком оригинальных идей, может проявить себя. Стоит также отметить, что идея плазмоидов понравилась не только создателям рабочего стола, и постепенно проникает и в другие области KDE-строения.

Как и все другие расширения KDE, с технической точки зрения плазмоиды представляют собой разделяемые библиотеки. Программирование плазмоидов – дело нетрудное, а вот их отладка – довольно нудный процесс. Когда мы отлаживаем обычные приложения, нам часто приходится перезапускать их; когда мы отлаживаем расширения Plasma, нам порой приходится перезапускать Плазму. Впрочем, и это нетрудно сделать – перезагружать X-сервер, по крайней мере, не придется. К тому же для отладки плазмоидов существует специальный инструмент – *plasmoidviewer*.

Покажите мне код!

Основой любого плазмоида является класс, наследующий Plasma::Applet или одному из его потомков. В библиотеке *libplasma* определены Plasma::GLApplet – специальный класс для создания плазмоидов, использующих OpenGL, и Plasma::PopupApplet – каркас для создания «всплывающих» плазмоидов. Все они происходят от QGraphicsWidget. От него же ведут свою родословную несколько классов, предназначенных для отображения данных внутри плазмоида: Plasma::Flash (вывод текста или графики в течение некоторого промежутка времени), Plasma::Icon (пиктограмма), Plasma::Meter (для отображения шкал уровня в различных форматах), Plasma::SignalPlotter (отрисовка нескольких графиков, меняющихся с течением времени) и Plasma::WebContent, который способен отображать содержимое web-страниц, заданных файлом или ссылкой.

Наш первый плазмоид распечатывает содержимое файла **/proc/cpuinfo** в своем окне (это не самое интересное, что можно сделать с помощью Plasma, но все не Hello World). Ниже приводится объявление класса плазмоида (полные исходные тексты вы, как всегда, найдете на диске).

```
class SysInfo_1 : public Plasma::Applet {
  Q_OBJECT
public:
  SysInfo_1(QObject *parent, const QVariantList &args);
  ~SysInfo_1();
  void paintInterface(QPainter *painter,
  const QStyleOptionGraphicsItem *option, const QRect&
contentsRect);
  void init();
  private:
  Plasma::Svg m_svg;
};
K_EXPORT_PLASMA_APPLET(sysinfo_1, SysInfo_1)
```

Как видно, он содержит объявления четырех основных методов,

Месяц назад Новые технологии KDE 4 и базовая структура обычной программы.

Plasma

которые обычно перекрываются в классах-потомках Plasma::Applet – конструктора, деструктора, paintInterface() и init(). Из них нам реально понадобятся только два – конструктор и paintInterface().

Макрос K_EXPORT_PLASMA_APPLET() выполняет рутинную работу по подготовке созданного нами класса к экспорту из библиотеки плазмоида. Первый его аргумент – имя библиотеки (без указания номера версии и расширения **.so**), второй – имя класса плазмоида. Ради удобства, а также для того, чтобы избежать некоторых неочевидных ошибок, я рекомендую использовать единую систему имен для всех файлов проекта плазмоида. Например, если целевая библиотека будет носить имя **sysinfo_1.x.so**, файлы с объявлением и определением класса плазмоида мы назовем **sysinfo_1.h** и **sysinfo_1.cpp**, соответственно. Файл описания плазмоида получит имя **sysinfo_1.desktop**.

Несколько слов о заголовочных файлах. В разных версиях *libplasma*, соответствующих различным версиям KDE 4, их имена могут варьироваться. В той системе, в которой я писал эти примеры, заголовочные файлы имеют вид: <plasma/applet.h>, <plasma/widgets/lineedit.h>. В вашей системе они могут выглядеть так же, а могут – и иначе, например: <Plasma/Applet>, <Plasma/Widgets/LineEdit>. CMake сам по себе тут не поможет, ведь он не станет править ваши исходные тексты [можно, правда, воспользоваться интроспекцией и условными директивами препроцессора, – *прим. ред.*]! Так что вам придется провести собственное исследование директории **/usr/include/**. Помните также, что интерфейс *libplasma* продолжает меняться. Примеры, подготовленные для этой статьи, используют некий общий знаменатель для всех версий API *libplasma* (начиная с KDE 4.0.*). Кроме того, в тексты примеров добавлены вызовы методов, определенных в более поздних версиях KDE. Эти вызовы закомментированы, и к ним даны пояснения. Если вы пользуетесь KDE 4.1 или KDE 4.2, можете раскомментировать их и сравнить результаты.

Как вы, конечно, поняли, самый интересный метод в классе SysInfo_1 – paintInterface(). Он выполняет отрисовку интерфейса плазмоида:

```
void SysInfo_1::paintInterface(QPainter *p, const
QStyleOptionGraphicsItem *option, const  QRect &contentsRect) {
  p->setRenderHint(QPainter::SmoothPixmapTransform);
  p->setRenderHint(QPainter::Antialiasing);
  m_svg.resize((int)contentsRect.width(), (int)contentsRect.height());
  m_svg.paint(p, (int)contentsRect.left(), (int)contentsRect.top());
  QFile file("/proc/cpuinfo");
  file.open(QIODevice::ReadOnly);
  QByteArray ba = file.readAll();
  file.close();
  QString s(ba);
  p->save();
  p->setPen(Qt::white);
  p->drawText(contentsRect, Qt::AlignTop | Qt::AlignLeft, s);
  p->restore();
}
```

Метод paintInterface() вызывается системой Plasma каждый раз, когда нужно перерисовать плазмоид. Помимо прочего, ему передается указатель на объект QPainter (переменная p) и ссылка на объект, обозначающий прямоугольную область вывода изображения, учитывающую отступы от края окна плазмоида (переменная contentsRect). В простейшем случае отрисовка интерфейса сводится к выводу графики с помощью объекта p.

В процессе отображения интерфейса плазмоида активно используются функции *libplasma*, предназначенные для работы с форматом SVG. Это неудивительно, если вспомнить, что плазмоиды можно масштабировать и вращать прямо на рабочем столе (разумеется, при работе с QPainter вы можете использовать любые графические функции, но для того, чтобы ваш плазмоид выглядел профессионально, следует предусмотреть возможность масштабирования и вращения). Помимо прочего, использование SVG позволяет настраивать внешний вид плазмоида, не внося изменений в его код, что должны высоко оценить авторы тем для оформления KDE 4. За работу с SVG в *libplasma* отвечает класс Plasma::Svg (мы сохраняем объект данного типа в переменной m_svg). Класс Plasma::Svg может не только преобразовывать данные из формата SVG в растровое изображение: он также «знает», где искать стандартные файлы изображений KDE, и может отрисовывать отдельные фрагменты файлов SVG, заданные своими идентификаторами. Последнее свойство очень полезно, так как позволяет хранить графические элементы сложного интерфейса приложения в одном файле, даже если они должны отображаться независимо друг от друга. Мы используем объект m_svg для отрисовки элементов окна плазмоида. Содержимое файла **/proc/cpuinfo** выводится на экран с помощью метода drawText() объекта QPainter. Хотя это и не обязательно, правила хорошего тона требуют вызвать метод save() объекта QPainter перед выводом собственной графики и метод restore() по окончании процесса.

Метод init() выполняется системой Plasma после вызова конструктора, причем только один раз. В нем можно поместить код, инициализирующий элементы плазмоида после того, как объект уже создан. Если в ходе этого процесса возникла ошибка, вы можете вызвать метод объекта-апплета setFailedToLaunch(). Первый его аргумент имеет тип bool (значение true указывает, что в процессе инициализации плазмоида произошла ошибка). Второй параметр типа «строка» позволяет сообщить системе, что же именно стряслось. Вызов setFailedToLaunch() – вежливый способ аварийного завершения плазмоида. В тот же файл, где мы храним реализации методов его класса, необходимо добавить строчку

```
#include "sysinfo_1.moc"
```

И что получилось?

Еще нам нужно создать файл описания плазмоида с расширением .desktop. Кстати, он требуется не только плазмоидам, но и вообще любым расширениям KDE. В .desktop- файле хранится описание расширения, а также инструкции для его загрузки системой. Наш файл расширения (**sysinfo_1.desktop**) выглядит так:

```
[Desktop Entry]
Name="Plasmoid Demo"
Comment="Plasmoid Demo"
Type=Service
```

```
X-KDE-ServiceTypes=Plasma/Applet
X-KDE-Library=sysinfo_1
X-KDE-PluginInfo-Author=Andrei Borovsky
X-KDE-PluginInfo-Email=anb@symmetrica.net
X-KDE-PluginInfo-Name=sysinfo_1
X-KDE-PluginInfo-Version=0.1
X-KDE-PluginInfo-Website=http://symmetrica.net
X-KDE-PluginInfo-Category=Examples
X-KDE-PluginInfo-Depends=
X-KDE-PluginInfo-License=GPL
X-KDE-PluginInfo-EnabledByDefault=true
```

Параметру Name присваивается имя расширения, предназначенное для показа пользователю. Под ним плазмоид будет известен, например, утилите добавления виджетов на рабочий стол. Параметр Comment содержит описание. Поле Type сообщает системе, что данное расширение относится к типу «сервис». Среди других важных параметров стоит отметить X-KDE-ServiceTypes, X-KDE-Library (имя библиотеки расширения) и X-KDE-PluginInfo-Name (имя плазмоида, используемое системой). Все остальное в данном файле – «лирика».

Нам остается только создать файл инструкций для CMake (см. стр. 92).

```
project(sysinfo_1)
find_package(KDE4 REQUIRED)
include(KDE4Defaults)
find_package(Plasma REQUIRED)
add_definitions (${QT_DEFINITIONS} ${KDE4_DEFINITIONS})
include_directories(${KDE4_INCLUDES} )
set(sysinfo_1_SRCS sysinfo_1.cpp)
kde4_add_plugin(sysinfo_1 ${sysinfo_1_SRCS})
target_link_libraries(sysinfo_1 ${PLASMA_LIBS} ${KDE4_KDEUI_LIBS})
install(TARGETS sysinfo_1 DESTINATION ${PLUGIN_INSTALL_DIR})
install(FILES sysinfo_1.desktop DESTINATION ${SERVICES_INSTALL_
DIR})
```

Новых для нас элементов в этом файле не так уж и много. Мы загружаем пакет Plasma, без которого плазмоид, естественно, компилироваться не может. Команда kde4_add_plugin() указывает, что целью сборки является расширение KDE 4 (а не исполняемый файл, как в прошлый раз). Теперь вы можете скомандовать

```
cmake <project_dir>
make
```

В результате в каталоге проекта появится поддиректория **lib**, в которой будет создан файл **sysinfo_1.so**. Далее можно было бы перейти в режим root и скомандовать

```
make install
```

› Рис. 1. Плазмоид в деле.

но тут нас поджидает одна неприятная неожиданность. Хотя в нашем проекте CMake и есть инструкции для генерации цели install в Make-файле, и соответствующий раздел будет создан, инсталляция плазмоида по умолчанию выполняется не туда, куда следует (этот эффект замечен и в KDE 4.0.x, и в KDE 4.1). Например, в моей системе библиотека **sysinfo_1.so** по умолчанию устанавливается в **/usr/local/lib/kde4/**, а файл **sysinfo_1.desktop** – в **/usr/local/share/kde4/services/**. Система не станет искать новый плазмоид в этих директориях, а значит, и не найдет. Правильные каталоги (для моей системы) – это **/usr/lib/kde4/** и **/usr/share/kde4/services/** соответственно. Чтобы заставить плазмоид работать, можно скопировать файлы **.so** и **.desktop** в указанные директории вручную, а можно попробовать запустить команду **cmake** с ключом -DCMAKE_INSTALL_PREFIX=/usr/lib/kde4/.

Увидеть установленный плазмоид в работе можно после перезапуска Plasma (или с помощью утилиты *plasmoidviewer*, о которой будет сказано ниже). Выполняется он очень просто – двумя командами из окна консоли от имени обычного пользователя:

```
kquitapp plasma
plasma
```

Не вздумайте закрывать окно консоли, скомандовав **kquitapp plasma**! В этом случае вы рискуете остаться один на один с серым экраном, и вот тогда уже для продолжения работы придется перезапускать систему. После рестарта Plasma апплет Add Widget должен «увидеть» новый плазмоид (**рис. 1**), а мы сможем добавить наше творение на рабочий стол и прочитать содержимое файла **/proc/cpuinfo** (**рис. 2**).

› Рис. 2. Добавление плазмоида на рабочий стол.

Управляемый виджет

Большинство плазмоидов KDE 4 делают практически то же, что и наш «первый блин» – просто показывают пользователю какие-то данные. Справедливости ради надо сказать, что плазмоиды редко отображают статичную информацию, как в нашем примере: чаще данные считываются из какого-то динамического источника с регулярными интервалами. KDE 4 предоставляет специальный механизм для работы с такими источниками данных (об этом мы поговорим в следующей статье). Сейчас же мы напишем еще

один простой плазмоид, но на сей раз добавим в него элементы управления, а точнее, один-единственный виджет – строку ввода. Плазмоид будет выполнять набранные в нем команды Linux (как это всегда бывает в учебных программах, функциональность не оригинальна, но реализация – поучительна). За основу в нашем примере взят плазмоид applet-runner, написанный Себастьяном Кжижковяком [Sebastian Krzyszkowiak].

Изначальный вариант плазмоида был рассчитан на KDE 4.0.x. Я добавил в него вызов некоторых методов, реализованных в более поздних версиях KDE 4, но на всякий случай закомментировал их, снабдив соответствующими пояснениями. Если вы работаете с более новой версией KDE 4 и хотите поэкспериментировать, можете раскомментировать эти вызовы.

При написании плазмоидов с элементами управления возникает соблазн воспользоваться стандартными виджетами из библиотек *Qt*/KDE, однако на данном этапе развития KDE 4 это может привести к краху системы. Помимо прочего, следует учесть, что для стандартного объекта-виджета трудно подобрать достойного «родителя». Главный класс плазмоида является потомком QGraphicsWidget, который не происходит от QWidget. QGraphicsWidget появился в библиотеке *Qt 4.2* как компромиссное решение, которое позднее утратило свою актуальность. С самого начала разработчики Graphics View Framework понимали, что программисты захотят встраивать виджеты в графические сцены, однако механизм для этого в *Qt 4.2* реализован не был (он появился в *Qt 4.4*). Вместо него разработчики ввели класс QGraphicsWidget, который, как и все классы-элементы графической сцены, является потомком класса QGraphicsItem, но при этом содержит ряд методов, подобных методам класса QWidget. Предполагалось, что класс QGraphicsWidget поможет программистам портировать виджеты на систему QGraphicsItem. Позднее разработчики пришли к более логичному решению – встраиванию уже готовых «настоящих» виджетов, потомков класса QWidget, но система Plasma, по крайней мере, в нынешнем своем воплощении, реализует более ранний подход.

Так или иначе, в настоящее время в апплетах для Plasma рекомендуется использовать виджеты, определенные в библиотеке *libplasma*: Plasma::CheckBox, Plasma::ComboBox, Plasma::GroupBox, Plasma::Label, Plasma::LineEdit, Plasma::TextEdit, Plasma::PushButton, Plasma::RadioButton. Все эти классы, в конечном счете, являются потомками класса QGrapihcsItem (а не QWidget), однако с виджетами, которые они реализуют, нельзя выполнять многие операции, допустимые при работе с объектами других классов-потомков QGrapihcsItem.

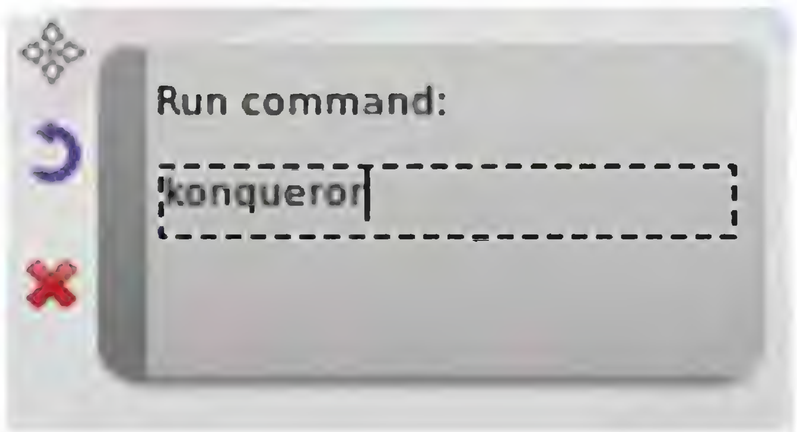
Наш плазмоид использует виджет Plasma::LineEdit для ввода команд. В моей системе этот виджет объявлен в заголовочном файле <plasma/widgets/lineedit.h>, а вашей он может быть объявлен в файле <Plasma/Widgets/LineEdit>. Объект класса Plasma::LineEdit создается в методе init() главного объекта плазмоида (наконец-то он нам пригодился!):

```
void PlasmaRunner::init() {
  pole = new Plasma::LineEdit(this);
  pole->setPlainText("");
  pole->setDefaultTextColor(QColor("black"));
  connect(pole, SIGNAL(editingFinished()), this, SLOT(run()));
}
```

Мы работаем с Plasma::LineEdit как с обычным виджетом – создаем объект, настраиваем его внешний вид, связываем сигнал объекта-виджета со слотом. В более поздних версиях KDE 4 у класса Plasma::LineEdit появляется метод setStyleSheet(), который позволяет настраивать внешний вид с помощью стандартного описания стиля. Если вы работаете в такой версии KDE 4, можете использовать его для придания строке ввода желаемого экстерьера, мы же воздержимся ради универсальности.

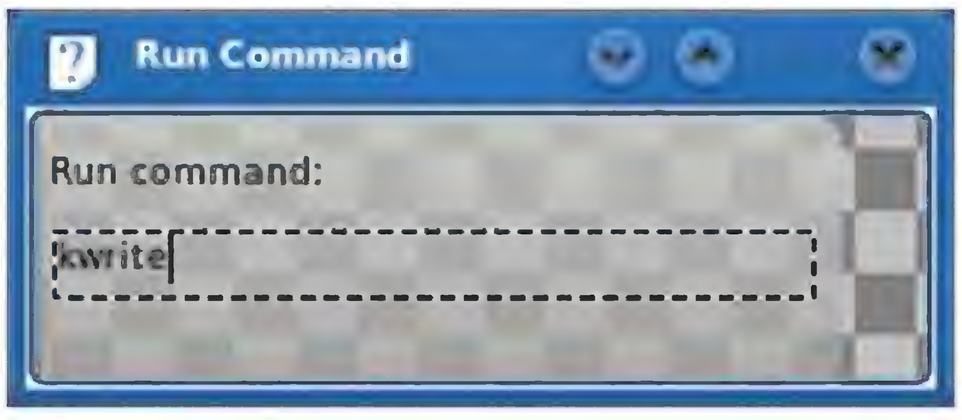
Метод paintInterface() плазмоида с элементами управления интересен тем, что объединяет отображение виджета с выводом графики (в нашем случае – поясняющей подписи) с помощью объекта QPainter:

```
void PlasmaRunner::paintInterface(QPainter *p,
      const QStyleOptionGraphicsItem *option, const QRect
&contentsRect) {
  p->setRenderHint(QPainter::SmoothPixmapTransform);
  p->setRenderHint(QPainter::Antialiasing);
  p->save();
  p->setPen(Qt::black);
  if (formFactor() == Plasma::Planar || formFactor() ==
Plasma::MediaCenter) {
  p->drawText(contentsRect, Qt::AlignTop, i18n("Run command:"));
  pole->setGeometry(QRectF(0, 20, contentsRect.width(),
contentsRect.height() ));
 } else {
  pole->setGeometry(QRectF(0, 5, contentsRect.width()
,contentsRect.height() ));
 }
  p->restore();
 }
```



› Рис. 3. Плазмоид с элементами управления.

Остальные элементы интерактивного плазмоида (**рис. 3**) похожи на рассмотренные ранее. Чтобы посмотреть, как выглядит и работает плазмоид, вовсе не обязательно перезапускать Plasma. Специальная утилита *plasmoidviewer* позволяет загрузить собранный и установленный плазмоид в обычном окне программы KDE (**рис. 4**). Перед запуском программы *plasmoidviewer* следует вызвать программу kbuildsycoca4, которая обновит кэш KDE 4. Аргументом *plasmoidviewer* должно быть имя плазмоида, но не то, которое отображается в апплете Add Widgets, а то, которое указано в параметре X-KDE-PluginInfo-Name .desktop-файла.



› Рис. 4. Утилита plasmoidviewer.

Как было сказано выше, стандартные виджеты *Qt* слабо интегрированы в систему плазмоидов. А можно ли использовать расширения *libplasma* в обычных *Qt*-программах вместе с системой Graphics View Framework? Теоретически – да, практически – нет, по крайней мере, пока. Объекты класса Plasma::SignalPlotter можно заставить работать в обычной программе; другие элементы *libplasma* либо не работают, либо приводят к краху приложения. Будем надеяться, что ситуация улучшится в будущих версиях KDE. LXF

» Через месяц Как получить данные для отображения на плазмоиде в режиме реального времени.

Пакеты расширений и новые версии

ЧАСТЬ 4 Завершая свой цикл статей об открытой вычислительной лаборатории, **Александр Бикмеев** научит, как нарастить возможности *Scilab*, и покажет, что еще есть в мире свободного математического ПО.

За три прошедших месяца мы с вами познакомились с некоторыми возможностями системы компьютерной алгебры *Scilab 4.1.2*. Мы занимались простыми вычислениями, создавали свои функции и модели для изучения физических процессов. В заключение мне хотелось бы кратко рассказать о пакетах расширений (Toolbox), новых функциях, появившихся в *Scilab 5.0,* и сообщить еще об одном проекте, не попавшем в Сравнение данного номера.

Расширения

Любой программный продукт должен развиваться, чтобы не отставать от требований времени. Если приложение небольшое и армия его поклонников насчитывает всего лишь сотни пользователей, а команда разработчиков трудится в тесном контакте друг с другом, то частый выход обновленных версий вполне может решить эту задачу. Однако такая ситуация встречается редко. Пользователи желают более частых обновлений, а создателям программы необходимо время на разработку и тестирование. Одним из гениальных решений этой вечной проблемы является механизм дополнений, расширяющих функциональность базового приложения.

Такие дополнения для систем численных расчетов обычно называют Toolbox [набор инструментов] или «пакет расширений». Для *Scilab* существует два вида подобных пакетов: расширения базовой системы и расширения системы визуального моделирования *Scicos*. Ссылки на те и другие можно найти в репозитории пакетов расширений (**http://www.scilab.org/contrib**), а также на официальном сайте *Scicos* **www.scicos.org**.

Пакеты *Scilab* разнородны по своей структуре. Это связано с тем, что они создавались на протяжении почти десяти лет существования системы: какие-то уже не поддерживаются, а какие-то только родились; однако в корневой директории любого пакета должны быть подкаталог **macros**, содержащий файлы исходных текстов его функций (расширение **.sci**) и/или их бинарные, то есть уже скомпилированные, версии (расширение **.bin**), и два главных файла: **builder.sce** и **loader.sce**, для сборки и загрузки пакета соответственно. Обычно в корневом каталоге пакета также присутствуют файлы **readme.txt** или **instal.txt**, в которых описывается процесс установки. Поскольку некоторые пакеты создаются на языке C/C++ или Fortran, то в корневом каталоге могут также присутствовать другие файлы, позволяющие собрать пакет стандартным способом: **./configure && make**. Мы же рассмотрим установку пакетов средствами самого *Scilab*.

Итак, предположим, вы скачали пакет OpenFEM (**http://www-rocq.inria.fr/OpenFEM**), который является инструментарием для твердотельного моделирования при помощи метода конечных элементов. Распакуйте его в какой-нибудь каталог, но помните, что данное место будет постоянным, то есть если вы впоследствии удалите эту директорию, то уничтожите вместе с ней и пакет OpenFEM, что может привести к нестабильной работе всей системы. Поэтому создайте в своем домашнем каталоге специальный подкаталог, в который будете помещать все устанавливаемые пакеты расширений, и назовите его, например, **ToolBoxes**. Затем распакуйте в него загруженный вами архив. Таким образом, пакет окажется в каталоге **/home/username/Toolboxes/openfem-scilab**.

Теперь нам необходимо скомпилировать пакет, то есть создать бинарные версии функций. Для этого запустите *Scilab* и выполните в главном окне команду (только не забудьте заменить username на ваше имя пользователя):

```
--> exec('/home/username/Toolboxes/openfem-scilab/builder.sce');
```

Если вы не поставите в конце точку с запятой, то весь процесс сборки будет отображаться на экране. Иногда это полезно, но чаще всего лишь засоряет консоль второстепенной информацией. По завершении сборки вновь появится приглашение консоли *Scilab*, и вы сможете загрузить пакет командой

```
--> exec('/home/username/Toolboxes/openfem-scilab/loader.sce');
```

Многие пакеты содержат документацию по каждой своей функции, и, что самое приятное, после их загрузки данная информация появляется в браузере справки. Чтобы проверить это, откройте окно справки *Scilab* (Help > Help browser) и перейдите в конец списка. Если сборка прошла успешно, вы увидите там три новых пункта, начинающихся с OpenFEM. Однако если вы скомпилировали пакет, но не подгрузили его, то к справке ничего не добавится. То есть, после закрытия и повторного запуска *Scilab* новые пункты исчезнут и не появятся до тех пор, пока вы опять не подгрузите данный модуль.

Важно запомнить, что, во-первых, нельзя изменять путь к скомпилированному пакету расширений, а во-вторых, необходимо выполнять команду загрузки пакета перед каждым использованием его функций. Если вы постоянно работаете с каким-либо пакетом расширений и не хотите постоянно вводить команду для его загрузки, то можете создать файл с перечнем команд, которые необходимо выполнить при запуске *Scilab*, и поместить в него команды загрузки всех необходимых вам пакетов расширений. Для этого откройте домашний каталог в *Konqueror* или *Nautilus*, включите отображение скрытых файлов (Вид > Показывать скрытые файлы) и перейдите в каталог **.Scilab/scilab-4.1.2**. Затем щелкните в пустой области окна правой кнопкой мыши и выберите пункт Создать > Текстовый файл. В появившемся диалоге введите имя файла **.scilab** и нажмите клавишу ENTER. Теперь введите команду загрузки необходимого вам пакета или пакетов, сохраните и закройте файл. В дальнейшем вы можете добавлять сюда любые команды, которые будут вам необходимы, и они будут выполняться сразу после запуска *Scilab* до того, как появится приглашение консоли.

Если пакет по каким-то причинам не собирается, то не завершайте команду его сборки символом «точка с запятой» – это поможет вам выявить проблемы. Предположим, вы скачали пакет ANN Toolbox 0.4.2 для работы с нейронными сетями. Скомпилировать его в текущем состоянии не получится – видимо, это связано с тем, что он был разработан еще для *Scilab* версии 2.6. При отсутствии точки с запятой в конце, команда сборки выдает множество сообщений с текстом: «...endfunction Missing...» – отсутствует служебное слово **endfunction**, которое закрывает описание функции. Чтобы привести пакет в рабочее состояние, достаточно войти в подкаталог **macros** и добавить в конец всех файлов с расширением **.sci** необходимое служебное слово. После этого пакет соберется без проблем.

Установка пакетов расширений системы визуального модулирования *Scicos* ничем не отличается от установки обычных пакетов самого *Scilab*, но их также можно собрать обычным способом, поскольку большинство из них представляют собой подпрограммы, написанные на C/C++, реже на Fortran. Однако по моему опыту использования MODNUM (**http://www.scicos.org/ScicosModNum/modnum_web/web/eng/eng.htm**), пакета для моделирования систем связи, лучше все-таки использовать уже готовые скомпилированные версии, которые достаточно лишь распаковать в какой-либо каталог и подгрузить обычным образом.

Месяц назад Имитационное моделирование в *Scicos*: через тернии к звездам.

Многие пакеты поставляются с файлами примеров или демонстраций, которые обычно расположены в каталогах **examples** и **demos** соответственно. Чтобы открыть их, следует выполнить ту же самую команду **exec** с указанием полного пути к файлу примера или демонстрации. Постоянное использование полной записи пути к файлу утомительно, поэтому вы можете просто сменить текущий каталог на каталог с примерами/демонстрациями. Делается это при помощи следующих команд.

```
--> SavedDir = pwd()
--> chdir('/home/username/Tolboxes/openfem-scilab/demos');
--> exec('gartfe.sce')
.....
--> chdir(SavedDir);
```

В первой строке мы запоминаем текущий рабочий каталог, во второй – меняем его на каталог с демонстрационными файлами пакета OpenFEM и выполняем демонстрационный файл. Затем мы можем выполнить еще несколько действий (это обозначено многоточием). Последняя команда позволяет вернуться к стандартному рабочему каталогу, который мы запомнили ранее.

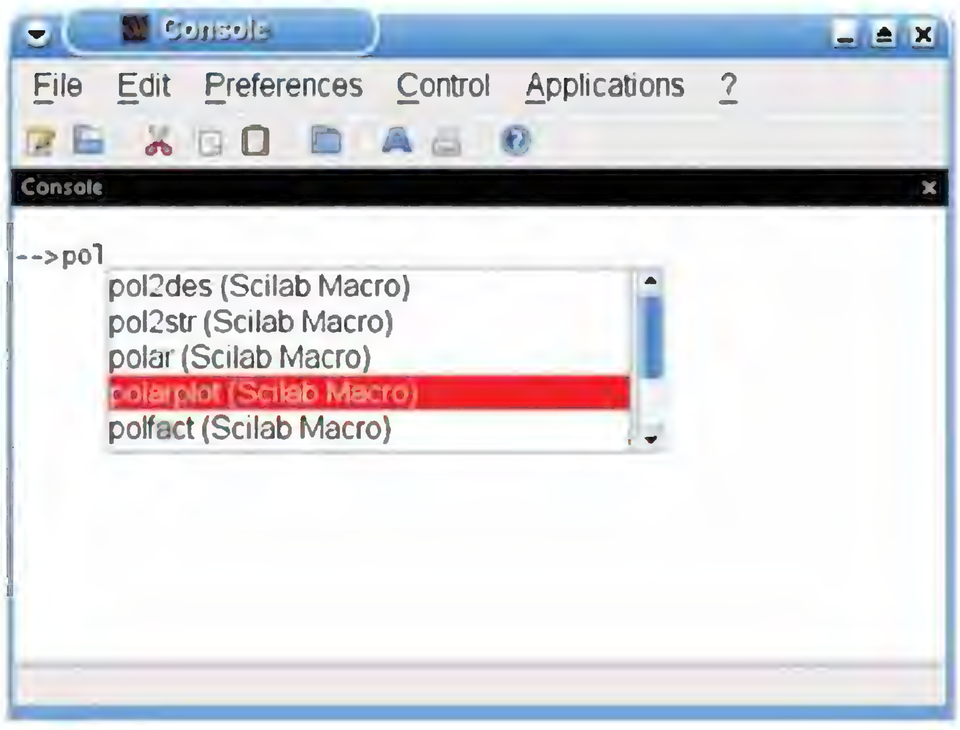
Пользуясь приведенной выше информацией, вы сможете найти, загрузить и установить необходимые вам пакеты расширений. Перечислять все имеющиеся варианты не имеет смысла, поскольку, во-первых, их достаточно много, а во-вторых, каждый пакет может содержать от 10 до 100 функций, что просто невозможно описать в рамках одной статьи. Поэтому подключайтесь к сети Интернет, заходите по указанным ссылкам или открывайте ваш любимый поисковик и ищите то, что вам нужно. А может быть вы чувствуете в себе силы создать новый Toolbox, или портировать какой-либо пакет из *GNU Octave* или *Matlab*? Действуйте, и пусть это принесет пользу не только вам, но и сообществу.

Scilab 5.0

В самой первой статье (**LXF106**) я говорил, что в мае 2008 разработчики обещают выпустить финальный релиз версии 5.0. Однако, как часто бывает со свободными (и не только свободными) проектами, этого не случилось. 23 июля на сайте появилась лишь версия 5.0-beta4. Она содержит хорошо известную ошибку графики, не позволяющую использовать ее на компьютерах с различными видеокартами. И хотя о ней было заявлено еще в апреле, разработчики до сих пор ее не исправили, поэтому поработать с графикой удастся не всем. Впрочем, вы можете установить на виртуальную машину Ubuntu Linux (при сборке из исходных текстов выявились определенные зависимости, позволившие предположить, что разработчики Scilab 5 используют Ubuntu, а может, и другой дистрибутив на базе Debian) и запустить двоичную версию. Именно так мне и удалось посмотреть все, что есть в 5.0-beta4. Можете пользоваться любым менеджером виртуальных машин; главное – это поиграть с настройками видеокарты и постараться выбрать более-менее старую модель с поддержкой OpenGL.

Прежде всего, изменилась лицензия *Scilab* – теперь он распространяется на условиях CeCILL, полностью совместимых с GPL v.2. Таким образом, *Scilab* стал полностью свободным приложением.

С точки зрения пользователя самым очевидным является изменение графического интерфейса. Он был переписан на Java Swing и стал более современным: появилась панель инструментов, изменились пункты меню, добавилась возможность настройки цвета фона и шрифта консоли Scilab (главное окно), улучшена работа с буфером обмена, в консоли теперь можно вводить комментарии и на русском языке. Кроме того, в консоли появилась возможность автоматического дополнения команд. То есть, вы можете ввести несколько букв, затем нажать клавишу Tab (Ctrl+Пробел), и на экране появится окно с вариантами продолжения, как показано на **рис. 1**.



› Рис. 1. Новый пользовательский интерфейс *Scilab 5.0* с окном автодополнения при вводе.

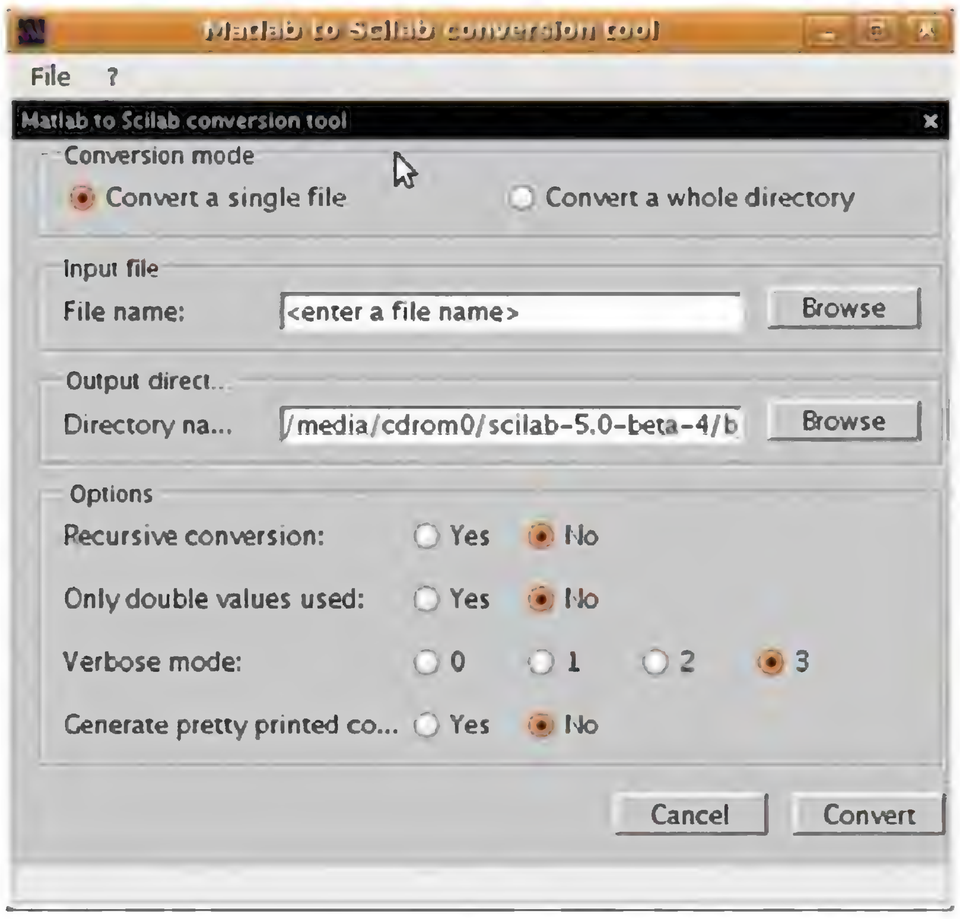
В появившемся меню Applications находятся пункты, при помощи которых можно вызвать дополнительные приложения:

» Редактор сценариев для создания файлов функций. С предыдущей версии *Scilab* он практически не изменился. В данном релизе не поддерживается режим отладки.

» Среду визуального моделирования систем *Scicos*, ту же версию, что и в *Scilab 4.1.2*.

» Графический редактор графов и сетей. Это новинка данной версии. Здесь вы можете нарисовать граф мышью и задать свойства его ребер и вершин, а затем просто выбрать меню Compute [Вычислить] чтобы найти кратчайший путь или выполнить другую операцию.

» Конвертор *Matlab-Scilab*. Этот новый инструмент представляет собой окно, в котором вы можете выбрать исходный и конечный файлы, а также настроить некоторые параметры преобразования. Более того, в качестве источника можно указать целую папку, что намного облегчает преобразование имеющихся пакетов расширений;



› Рис. 2. Окно инструмента для конвертирования файлов *Matlab* в синтаксис *Scilab*.

» Редактор переменных. Здесь вы можете просматривать описанные переменные данного сеанса, изменять их и удалять. Список переменных можно фильтровать по типу данных и/или шаблону имени. Редактор очень удобен и практически незаменим в процессе разработки приложений.

Улучшен также встроенный инструмент просмотра справочной информации, изменился внешний вид, добавлена возможность отправки текста справки на печать. Переработан и сам справочный материал – улучшена структура, появился раздел User's Guide [Руководство пользователя]. Впрочем, не все разделы справки еще заполнены, в некоторых местах вместо текста находится либо краткий перечень функций раздела, либо фраза TODO [сделать]. Так что справка еще будет дорабатываться, а со стороны российского пользователя хотелось бы увидеть русифицированную версию. Желающие поработать есть? Кстати, пункт запуска демонстраций переехал в меню справки и обзавелся новым интерфейсом в виде нескольких раскрывающихся списков, что сделало навигацию по демонстрационным примерам намного легче и нагляднее, да и набор примеров расширился.

Реализована новая архитектура *Scilab*, которая должна позволить обновлять систему и пакеты расширений при помощи патчей, тогда как

все предыдущие версии просто переустанавливались. Обновился движок отрисовки 3D – теперь он основан на JOGL, и поверхности действительно выглядят не такими угловатыми, как раньше.

Набор пакетов расширений (Toolbox), входящих в стандартную поставку, был расширен. Появились: *FFTW* (быстрое преобразование Фурье), *Genetic Algorithms* (работа с генетическими алгоритмами), *Metanet: Graph and Network* (пакет для работы с графами и сетями), *Simulated Annealing* (пакет оптимизации моделей), пакеты для работы с разреженными и многомерными матрицами, для экпорта результатов в TeX – и многие другие. Добавлены новые и переписаны некоторые старые функции для управления оболочкой, историей, вводом и выводом информации в/из файлов. Таким образом, функциональность стандартной версии пакета значительно расширилась.

Многим пользователям, которые не очень-то дружат с английским языком, может быть, полегчает, поскольку в текущей SVN-ветке я обнаружил файлы русской локализации оболочки. Более того, интерфейс версии 5.0-beta4 под Windows уже русифицирован. Почему этого не случилось с версией для Linux, я сказать не могу.

Учитывая такое количество нововведений и улучшений, наряду с хорошей оптимизацией расчетных алгоритмов (см. врезку), становится понятно, почему версия 5.0 ожидается с таким нетерпением.

FreeMat

В Сравнении данного номера не был указан еще один из лидеров среди *Matlab*-подобных пакетов – *FreeMat*, не говоря уже о множестве мелких программ, которые в большинстве своем предназначены для выполнения определенных узких задач; например, *Gpredict* (http://groundstation.sf.net/gpredict/) – модульное приложение для отслеживания и прогнозирования орбит спутников в реальном времени. А также пусть и коммерческий, но достаточно хорошо известный и популярный за рубежом пакет MuPAD (www.mupad.com). Но будем последовательны и рассмотрим лишь свободный вариант.

FreeMat не является полным клоном *Matlab*. Как указывается в FAQ на официальном сайте приложения (http://freemat.sf.net), этот пакет поддерживает приблизительно 95% функционала *Matlab*, что, согласитесь, немало. В пакете есть полноценная IDE с возможностью создания и отладки кода, сохранения и выполнения истории команд, управления файлами. Последний стабильная версия 3.6 была выпущена в марте этого года и доступна практически для всех известных платформ. Среди ее основных функций хочется отметить следующие:

- Поддержка объектно-ориентированного программирования и перегрузки операторов. То есть *FreeMat* – это единственный пакет численных вычислений, способный работать с файлами в новом синтаксисе *Matlab*.
- Поддержка автодополнения в консоли *FreeMat*.
- Поддержка параллельных вычислений на базе технологии MPI.
- Использование OpenGL для построения графиков и поверхностей.
- Предоставление большинства известных методов численных вычислений, включая прямое и обратное преобразования Фурье (FFT), преобразование Гильберта; SVD-, LU- и QR-разложения матриц, работу с многомерными матрицами, интерполяцию и аппроксимацию функций и многое другое.
- Прекрасное справочное руководство (на английском языке).

Самое удивительное в том, что этот проект, как указано на сайте, создан и развивается силами всего лишь одного человека – Самита Басу [Samit Basu], но, как он сам признается, ему помогали и помогают многие сторонние добровольцы (в частности, многие Toolbox'ы являются просто преобразованными версиями *GNU Octave*). Целью данного проекта является предоставление простого интерфейса для внешних программ на C/C++/Fortran, а также реализация свободной альтернативы *Matlab* «с человеческим лицом». Последнее замечание требует пояснения. Как признается сам автор, он использовал *Matlab* еще с версии 4.0 и всегда не мог понять, почему некоторые простые вещи сделаны вычурно и неоптимально, а при смене версий старые функции не переписываются с учетом новых алгоритмов и технологий. Впрочем, этот вопрос часто задают пользователи-профессионалы любого приложения. С другой стороны, именно он часто и приводит к появлению альтернатив с открытым кодом, что не может не радовать остальных людей.

На официальном сайте имеется очень слабый Wiki, FAQ, ссылки на пакеты для различных дистрибутивов и файлы документации в PDF-формате, а также блог разработчика. Часть последних сообщений в блоге посвящена готовящейся четвертой версии пакета. Основными изменениями будут:

- внедрение JIT-компилятора;
- система типов, совместимая с *Matlab*;
- новая система сборки проектов;
- динамическая компоновка библиотек BLAS.

Несмотря на провал в производительности (см. врезку), *FreeMat* также может рассматриваться как некая замена проприетарной и очень дорогой системе *Matlab*. Хорошая совместимость и схожесть интерфейса позволяет так построить процесс обучения студентов, что переход к лицензионным версиям *Matlab* на производстве будет практически незаметен.

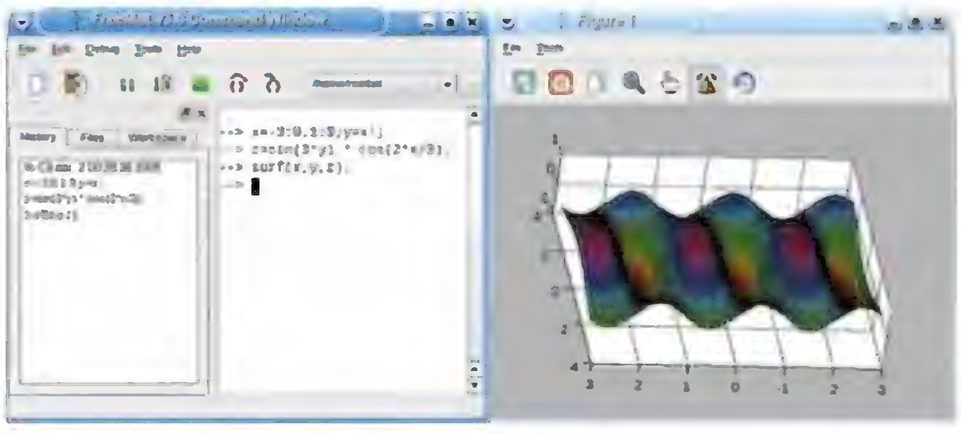


Рис. 3. IDE *Freemat* и окно с примером построения поверхности.

Пришла пора прощаться

Итак, мы завершаем данный цикл статей. Надеюсь, что он был вам полезен. Теперь дело за вами. Возьмите пакеты из Сравнения этого номера, скачайте новые версии *Scilab* и *FreeMat* и выясните, что больше всего подходит вам для того, чтобы начать или продолжить свои исследования при помощи свободного программного обеспечения. И даже если впоследствии вам понадобится *Matlab*, то переход не будет таким уже страшным. Но действуйте немедленно, помните, Нобелевская премия вручается каждый год, а вот ждать ее приходится десятилетиями, поэтому чем быстрее вы закончите эпохальное исследование, тем больше шансы, что вы доживете до вашей лекции в Стокгольме.

Руководители курсовых и дипломных работ могут привлечь студентов к разработке и/или портированию различных пакетов расширений для *Scilab* или другого пакета с открытым кодом. Если же вы студент и вас заинтересовал какой-либо проект, то обратитесь сами к своему руководителю или преподавателям на кафедре – может быть, кто-то возьмется вам помочь, и в итоге получится неплохая курсовая или дипломная работа. LXF

Ваше мнение

Что-то осталось для вас непонятным? Возникли какие-то трудности в работе, вопросы по использованию – а может, пожелания о дальнейшем развитии данного цикла? Напишите нам (letters@linuxformat.ru), и мы постараемся вам помочь.

Производительность

Мы провели те же самые тесты, что и в рубрике Сравнение данного номера: SVD-разложение и быстрое преобразование Фурье (FFT) для матрицы случайных действительных чисел размера 500x500. Тестирование проводилось на ноутбуке ASUS F3Ka с процессором Turion x2 TL-58 и 2 ГБ ОЗУ, под управлением Mandriva 2008.1 PP. Каждый тест был проведен 100 раз, и в качестве результата было взято среднее значение.

- *Scilab 5.0-beta4* показывает, что разработчикам действительно удалось повысить быстродействие, как они и обещали. Тест SVD показал увеличение скорости расчета на 14%, а FFT почти на 78%, то есть быстрое преобразование Фурье теперь выполняется почти в 4 раза быстрее.
- *FreeMat* не на высоте. В SVD-тесте с теми же параметрами, что и в Сравнении этого номера, он показал скорость в 1,7 раза меньше, чем *Scilab 4.1.2*. Так что с точки зрения производительности *FreeMat* нуждается в доработке. FFT-тест, к сожалению, провести не удалось вследствие неполной сборки пакета.

22-25 октября, 2008

МОСКВА, МВЦ "КРОКУС ЭКСПО"

ИНФОКОМ'08

ВРЕМЯ ВЫСОКИХ ТЕХНОЛОГИЙ

VIII МЕЖДУНАРОДНАЯ ВЫСТАВКА-ФОРУМ ИНФОКОММУНИКАЦИОННЫХ ТЕХНОЛОГИЙ

Разделы выставки:

- Аппаратные средства
- Программное обеспечение
- Системная интеграция
- Информационная безопасность
- Услуги по разработке ПО
- Фиксированная связь
- Мобильная связь
- Цифровое телерадиовещание
- Инфокоммуникационные услуги
- Почтовая связь
- ИКТ в национальных проектах
- Технопарки, Инновации
- Электронное правительство
- Электронные регионы
- Национальные экспозиции

Организатор:

FORMIKA

Тел.: +7 (495) 660 75 90
Факс: +7 (495) 660 75 89
www.infocomtech.ru

При поддержке:

МИНИСТЕРСТВО СВЯЗИ И МАССОВЫХ КОММУНИКАЦИЙ РОССИЙСКОЙ ФЕДЕРАЦИИ

Информационные спонсоры:

Информационные партнеры:

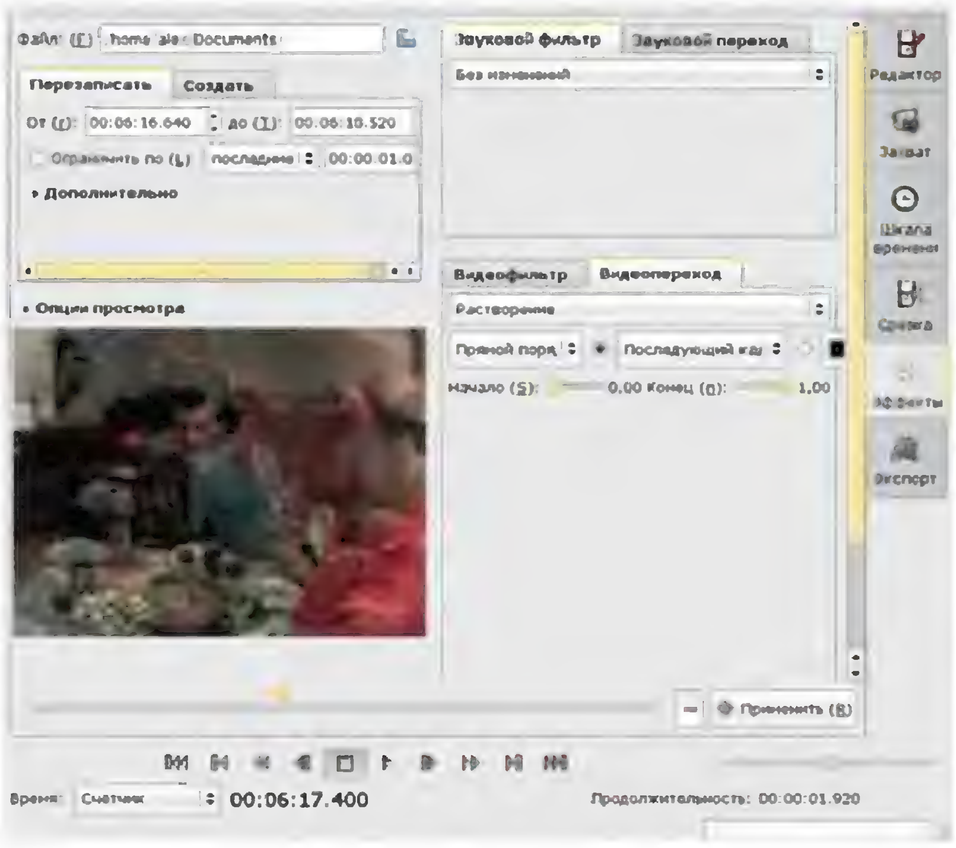
Создаем

ЧАСТЬ 3 Ничто не раздражает **Алексея Маслия** больше, чем резкие смены плана в любительской видеосъемке. К счастью, он знает, как это исправить.

Наш эксперт

Алексей Маслий
Познакомился с Linux в конце 1997г., а с 2003 использует его для решения любых задач, в том числе и для обработки видео.

На мой взгляд, наиболее неприятной чертой любительского видео является скачкообразный переход от одной сцены к другой. Видимо, нагрузка на мозг в таких случаях резко возрастает. Профессионалы уже давно нашли способ избавиться от данной проблемы: они создают видеопереходы. Наиболее распространенным эффектом (по крайней мере, в кино) является плавное уменьшение яркости до нуля в течение секунды или двух, после чего включается новая сцена. Современное профессиональное оборудование и программы содержат сотни всевозможных переходов. Ну, а что же мы, любители? У *Kino*, конечно, ассортимент не столь обширный (хотя список доступных эффектов растет от версии к версии), но для наших целей его вполне хватит. Лично я в своих фильмах вообще использую только два перехода: один – от сюжета к сюжету (под «сюжетом» здесь понимается набор сцен, объединенных заставкой, скажем, «В гостях у ...»), а другой – внутри сюжета между отдельными сценами. Мой выбор – это эффекты **Растворение** и **Вытеснение**; их мы сегодня и будем использовать в качестве примера. На вкус и цвет товарища нет – если вам ближе что-то другое, рецепты придется слегка скорректировать (например, заменить названия пунктов меню); принцип работы, конечно, не изменится.



› Рис. 1. Настройте видеопереход по своему вкусу.

Растворяй и властвуй

Запустите *Kino*, если вы еще этого не сделали, и откройте вкладку **Видеопереход**. В выпадающем меню следует выбрать пункт **Растворение**. Чуть ниже расположены еще два списка. Первый предлагает установить прямой или обратный порядок наложения эффекта. Вас честно предупредят, что последнее работает не всегда, так что оставим **Прямой порядок**. Второй список задает поведение следующей сцены. Выбор осуществляется между пунктами **Последующие кадры** или **Последующий кадр**. Разница в том, что в случае а) движущаяся картинка сменяется движущейся, а в б) – статической картинкой (первым кадром сцены). В качестве альтернативы можно указать однородный фон и цвет заливки. Обязательно следует задать временной интервал применения перехода: по умолчанию эффект накладывается на всю выбранную сцену, что в большинстве случаев не нужно (исключением является ситуация, когда сцена была специально создана для перехода). Поэтому на вкладке **Перезаписать** поправьте значения в полях **от** и **до** (по умолчанию они соответствуют положению начала и конца сцены во всем временном потоке фильма). Вряд ли разумно создавать переход длительностью больше двух секунд: я лично использую одну или меньше. Аналогичного результата можно добиться, перемещая ползунки **Начало** и **Конец**, расположенные под выпадающими меню, но пользоваться ими крайне неудобно: ввести время действия эффекта вручную существенно проще.

Скорая помощь

Эффект **Растворение** не обязан связывать две сцены. Его можно использовать также и для завершения сцены или сюжета плавным переходом от полноцветной картинки к статическому одноцветному фону (обычно черному). Я, как правило, применяю его в начале (после заставки) и в конце сюжета.

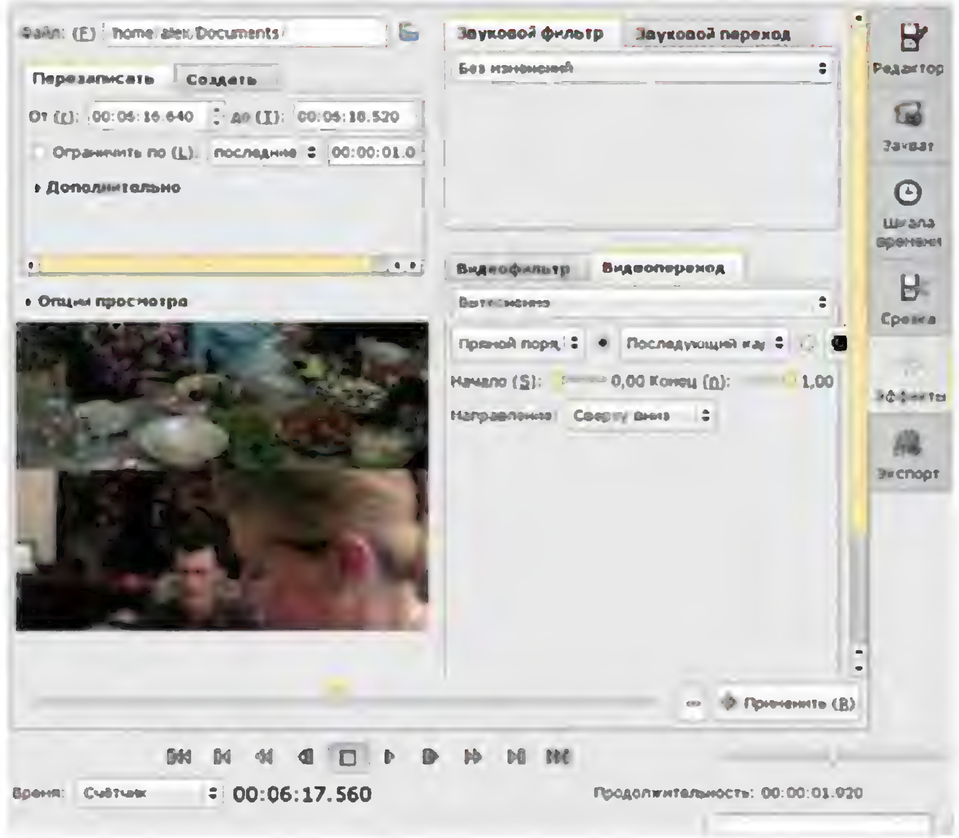
Итак, все готово (**рис. 1**). Еще раз хочу предупредить: обязательно оцените (или хотя бы ускоренно промотайте) результат в окне предварительного просмотра; нажатие кнопки **Применить** *необратимо*. На моей памяти, *Kino* несколько раз сбрасывал значения, введенные в поля **от** и **до**; было ли это следствием моих неправильных действий или ошибки в программе – сказать не могу. Кроме того, вы вполне можете (случайно или намеренно) ввести в поле **от** значение, превышающее **до**. К сожалению, такие вещи программа не отлавливает; результат непредсказуем и, как уже упоминалось выше, необратим.

Если же все в порядке и предпросмотр вас удовлетворяет, нажмите кнопку **Применить** и подождите несколько секунд, пока будет генерироваться новая сцена. Программа отрежет от выбранной сцены «хвост» заданной длительности, наложит на него эффект и вставит в виде новой сцены между двумя выделенными. Для присоединения отрезанного кусочка обратно очень удобно пользоваться механизмом объединения сцен.

Видеопереход **Вытеснение** я обычно использую для перехода между сценами внутри одного сюжета. Суть эффекта состоит в том, что новая сцена как бы выталкивает старую. Процедура работы с данным переходом похожа на рассмотренную выше, только в конце добавляется еще одно выпадающее меню (**рис. 2**). Оно определяет направление вытеснения: **Справа налево** или **Сверху вниз**. Лично мне больше по душе второе. Не забудьте указать временной интервал действия перехода и посмотреть, что получилось в итоге.

Месяц назад Мы монтировали отснятый материал и добавляли заставки.

ВИДЕОПЕРЕХОДЫ



› Рис. 2. Вытеснение, в сущности, ничем не отличается от Растворения.

Еще один практический совет: по завершении наложения всех эффектов переключайтесь в режим **Редактор** и склеивайте сюжет в одну сцену – это избавит вас от лишних сложностей в будущем. Если в дальнейшем предполагается производить над фильмом какую-либо дополнительную обработку, сохраняйте сюжет в виде отдельного проекта *Kino*.

Экспорт

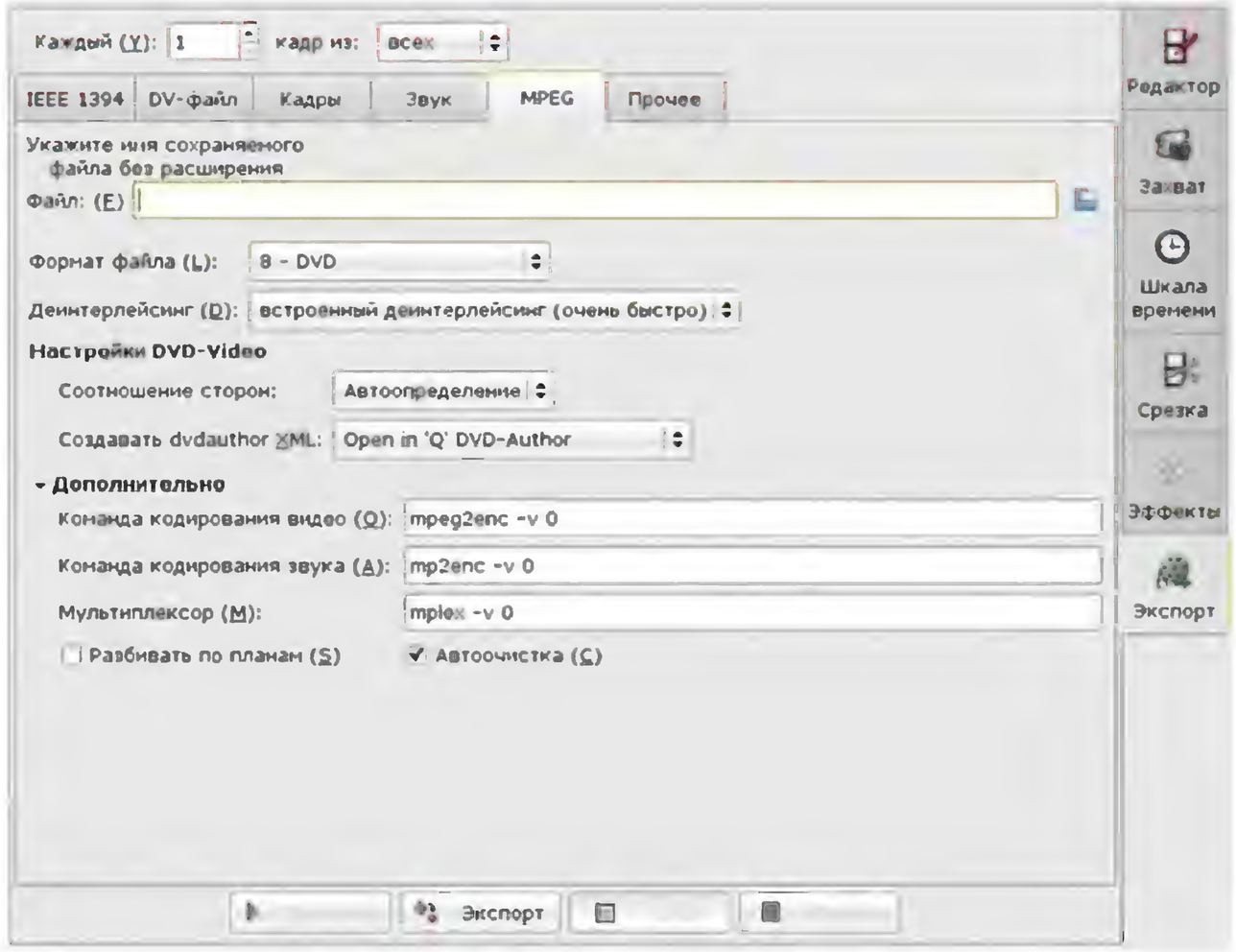
Итак, все эффекты наложены, переходы готовы. Пора браться за кодирование видео в формат, подходящий для DVD-плейеров. Это и многое другое можно сделать, активировав вкладку **Экспорт**. Центральная часть окна программы *Kino* примет вид, показанный на **рис. 3**. Рассмотрим каждую вкладку отдельно.

» **IEEE1394** Позволяет загрузить подготовленный фильм на цифровое устройство с DV-входом (напрмер, MiniDV-камеру).

» **DV-файл** Как нетрудно догадаться, в этом случае видео просто экспортируется в файл в формате DV. Вариации формата и параметры можно указать с помощью расположенных на вкладке органов управления. Данная опция будет полезна, если вы собираетесь обрабатывать смонтированный фильм в других программах, например, *Cinelerra*: наш журнал писал о ней в выпусках **LXF93–96**.

» **Кадр** Отсюда можно сохранять отдельные кары. Диапазон доступных форматов ограничен возможностями библиотеки *gdk-pixbuf* и включает PPM, JPEG, PNG, TIFF, GIF, BMP, TGA. Номер кадра к выбранному имени файла добавляется программой автоматически. Эта функция приходит на помощь, когда вы хотите выбрать подходящий кадр для заставки, но никак не можете поймать его с помощью кнопки **Стоп-кадр**. Не забудьте ограничить диапазон экспорта, иначе разобраться в сгенерированной куче файлов будет непросто.

» **Звук** Здесь можно экспортировать аудиопоток выбранной сцены или всего фильма в файл в формате WAV, MP2, MP3 или OGG. Воспользуйтесь данной функцией, если хотите обработать его во внешней программе (например, *Audacity* из **LXF106**; см. также **LXF99**).



› Рис. 3. Мы готовы экспортировать видео в формат, понимаемый DVD-проигрывателем.

Интересующие нас функции экспорта готового фильма в формат MPEG2 скрываются, как ни странно, сразу в двух вкладках: **MPEG** и **Прочее**. Первая содержит настроенные профили для экспорта фильма (то есть видео- и аудио-потоков) средствами пакета *mjpegtools*. В выпадающем меню предлагается выбрать из MPEG1, VCD, MPEG2, SVCD и DVD. Вкладка **Прочее** позволяет кодировать в форматы DVD, H.264, MP3, MPEG-4 (различные варианты), OGG, VCD и XviD при помощи очень быстрого кодека *ffmpeg*.

Давайте разберемся, как работать с каждым из них в отдельности. Начнем с *mjpegtools*, т.е. вкладки **MPEG**.

Убедитесь, что в самой верхней строке выбрана опция **Каждый 1 кадр из всех**. Можно кодировать либо с пропуском кадров, либо не весь фильм, а только отрезок из него (это удобно при нарезании материала на части). В строке **Файл** следует указать имя без расширения (можно также выбрать каталог, нажав на значок папки рядом с полем ввода). Далее, задайте формат кодирования (в нашем случае, DVD), активируйте встроенный деинтерлейсинг, прикажите создавать XML-файл для *QDVDAuthor* и, при необходимости, уточните опции кодека в разделе **Дополнительно** (**LXF102**). Кодирование аудио- и видеопотоков в *mjpegtools* производится двумя различными программами. При этом создаются два файла: **.mpv** – видеопоток, **.mp2** – аудиопоток, а по завершении процесса они сводятся воедино (данный процесс называется мультиплексированием). Галочка в пункте **Автоочистка** заставляет удалять файлы аудио- и видео-потоков по завершении мультиплексирования, вне зависимости от того, удачно оно прошло или нет. Обратите внимание – это означает, что реально на диске требуется вдвое больше места, чем потом будет занимать фильм.

Теперь все готово к нажатию кнопки **Экспорт**. После этого остается только ждать и следить за статусной строкой и индикатором процесса в правом нижнем углу экрана. Кодирование занимает довольно много времени, так что можете пока заняться чем-нибудь другим. По завершении процесса вы получите файл MPEG2 с готовым фильмом.

Работа с *Kino* окончена, и можно переходить к мастерингу диска DVD-Video.

Вкладка «**Прочее**» содержит всего два выпадающих меню. В первом предлагается указать формат кодирования, во втором – предустроенный профиль. Нас, естественно, будет интересовать DVD. Программа предлагает выбрать между одно- или двухпроходным кодированием. Первое всегда выполняется с постоянным битрейтом (CBR), а второе – с переменным (VBR). При прочих равных условиях, я советую выбирать двухпроходное кодирование – размер итогового файла будет заметно меньше. В выпадающем меню **Профиль** можно выбрать между обычным VOB-файлом и каталогом DVD-Video. Поскольку DVD-диск мы собираемся делать сами, то логичнее будет взять VOB (кстати, это стандартный файл MPEG2).

Снова жмем на кнопку **Экспорт**, не забывая, однако, проверить настройки временного диапазона: мы должны кодировать все. В отличие от *mjpegtools*, *ffmpeg* обрабатывает и аудио-, и видео-потоки одновременно, т.е. не столь требователен к дисковому пространству.

У вас может возникнуть резонный вопрос: какой же из двух кодеков выбрать? Для сравнения проведем небольшой тест. Возьмем DV-файл объемом 519 570 КБ и длительностью 2 минуты 28 секунд. Сначала запустим кодирование при помощи *mjpegtools* с настройками по умолчанию и включенным быстрым деинтерлейсингом. На моей машине это заняло 2 минуты 40 секунд, а результирующий файл «потянул» на 141 592 КБ (а значит, на диске требовалось дополнительно еще 140 МБ).

Теперь попробуем кодек *ffmpeg* с двухпроходным кодированием в VOB-файл. Это займет 3 минуты 20 секунд; в результате получится файл объемом 113 962 КБ. При просмотре не было замечено отличий ни в качестве видео, ни в качестве аудио (конечно, все это чисто субъективно). Какой же вывод? Кодек *mjpegtools* на 20% быстрее, но при этом создает файлы размером на 20% больше, чем *ffmpeg*. Однако *mjpegtools* можно дополнительно подстраивать (**LXF102**), и это идет ему в плюс. Конечно, кодек *ffmpeg* также имеет много опций, но на изучение всех аргументов командной строки потребуется немало времени.

Кстати, может показаться, что *Kino* не предоставляет возможности задействовать кодек *ffmpeg* с параметрами пользователя; однако это не так. На вкладке **MPEG** в разделе **Дополнительно** можно не только указать опции кодека, но и заменить *mjpegtools* на другой – тот же *ffmpeg*. Однако повторюсь – подобрать необходимые параметры для *ffmpeg* довольно трудно, особенно человеку новому или тому, чья основная деятельность никак не связана с производством и кодированием видео. В итоге при использовании стандартных настроек я бы выбрал *ffmpeg* (т.е. вкладку **Прочее**) за меньший размер итогового файла, а когда нужно что-то «подкрутить», то *mjpegtools*.

По завершении кодирования и решении всех связанных с этим вопросов мы получаем готовый фильм в виде файла (или набора файлов) в формате MPEG2. На этом работа с *Kino* заканчивается. Для создания диска DVD-Video требуется программа мастеринга DVD. Выбирая из имеющихся под Linux, я остановился на *QDVDAuthor*. LXF

А где взять?

Практически все программы, упомянутые на страницах этого учебника (кроме, пожалуй, *GIMP* и, в некоторых случаях, *Kino*) имеют одну общую черту: они не включаются на стандартный дистрибутивный носитель (если, конечно, у вас не Debian на 14 DVD), и их нужно доустанавливать из репозитория в Интернете. Если же такой возможности нет или вы просто хотите по-быстрому попробовать свои силы, я рекомендую применять специализированные «живые» дистрибутивы: Dyne:bolic (dynebolic.org), 64Studio (www.64studio.com) и Ubuntu Studio (ubuntustudio.org). Некоторые из них появлялись и на **LXF**DVD. Использование этих систем также поможет избавить ваших друзей от Windows-зависимости (**LXF102**). К сожалению, пока только Dyne:bolic содержит в своем составе программу DVD-мастеринга – *DVDStyler*. В качестве одной из ее приятных особенностей следует отметить прекрасную русификацию без дополнительных усилий со стороны пользователя. Кроме того, Dyne:bolic имеет просто беспрецедентную по простоте процедуру инсталляции, не требующую от вас ни переразбивки диска, ни даже перезагрузки компьютера (**LXF106**).

Один или несколько?

Кроме экспорта фильма в один MPEG2-файл, есть еще один вариант: использовать несколько файлов, каждый из которых будет содержать отдельный сюжет или часть сюжета. Недостаток данного подхода в том, что в куче роликов немудрено и заблудиться. Кроме того, работа по монтажу и склейке никуда не девается, просто выполнять ее приходится в другой программе. Ну и если вы задумали крутые переходы между сюжетами, то и в этом случае, конечно, все возможно, но придется немного попотеть.

Теперь о плюсах – их тоже немало. Задача разбивается на кусочки меньшего размера, и, следовательно, у вас меньше шансов запутаться и что-то испортить, чем в большом проекте. Кодирование частей фильма занимает, естественно, меньше времени, а освободить компьютер на час гораздо проще, чем на три часа. Кроме того, если при просмотре после кодирования вы замечаете ошибку (неправильно наложенные титры или переходы) или вам просто что-то не нравится, то на ее исправление опять-таки уходит гораздо меньше времени. Необходимость перекодирования может возникнуть и при желании немного «ужать» видео-файл (**LXF102**). Ну и, наконец, если у вас недостаток места на жестком диске, захват и обработка небольших сюжетов – это фактически единственный вариант.

» **Через месяц** Видео смонтировано и записано в подходящем формате: осталось прожечь DVD.

Мощность.
Надежность.
Производительность.

ETechnologies ETEGRO

СЕРВЕРЫ. СИСТЕМЫ ХРАНЕНИЯ ДАННЫХ.
ГРАФИЧЕСКИЕ СТАНЦИИ

Компания ETegro Technologies -
производитель системного программного
обеспечения и аппаратных решений:
серверов, графических станций,
кластеров и систем хранения данных.

Продукция компании ETegro
сертифицирована для работы под
управлением операционных систем
семейства Linux, что позволяет сократить
совокупную стоимость владения и
избежать дорогостоящих лицензионных
платежей. Признанием качества и
надежности серверов ETegro служит
тот факт, что компания стала первым
в России OEM-партнером Novell, Inc.

Более подробную информацию
о компании ETegro Technologies
можно получить на сайте www.etegro.com

Гарантии сохранности ваших данных.

Кстати.

Сравните цены с аналогами.

Центральный офис
Москва, Электродная ул., д. 2, стр. 12-13-14
Телефон: +7 (495) 380-02-88
Факс: +7 (495) 380-02-88

E-mail: sales@etegro.com
www.etegro.com

Реклама

Собираясь

ЧАСТЬ 1 Система сборки – неизменный спутник любого разработчика. К счастью, сегодня уже не обязательно вникать во все тонкости M4 – **Андрей Боровский** представляет CMake!

Вот вам мой вариант теста для настоящего программиста: что неприятнее всего не найти? Кредитку, которой только что расплачивался в ресторане, телефон прекрасной девушки, с которой вчера познакомился в метро, или файл сборки проекта для той системы, в которой работаешь? Разумеется, ни одна из этих проблем не стоит того, чтобы делать из нее трагедию. Даже если кто-то сумеет воспользоваться вашей карточкой, деньги, скорее всего, удастся вернуть, потеря случайного телефонного номера тоже не повод для грусти – кто знает, может оно и к лучшему, а отсутствующий в исходниках файл можно восстановить, при условии, что вы понимаете структуру проекта и у вас под рукой есть подходящие утилиты.

Традиционно программисты, работающие в Linux (да и других Unix-системах) используют для сборки программ инструментарий GNU build system или Autotools (состоящий из утилит *autoconf*, *automake*, *libtool*, *gnulib*). Именно с его помощью создаются знакомые всем нам сценарии *configure*, которые и генерируют make-файлы для сборки приложения. Система GNU build system настолько тесно связана с историей Linux и других открытых проектов, что «мантра» **configure && make && make install** рассматривается некоторыми Unix-программистами как единственно верный способ установки ПО. К недостаткам Autotools можно отнести ее ориентацию на инструментарий разработчика GNU (GNU make, *GCC* и т.п.), который очень популярен на открытых Unix-платформах, но мало распространен в других средах. Стремясь заполнить этот пробел, разработчики из компании Kitware создали свой собственный вариант кросс-платформенного генератора сборочных файлов – CMake (Cross-platform Make).

Система CMake выбрана в качестве стандартного средства сборки KDE 4, и уже по одной этой причине заслуживает нашего внимания. Но у нее есть и собственные достоинства. Прежде всего, CMake является по-настоящему кросс-платформенным генератором проектов, позволяющим создавать их единые описания для Linux, других Unix-систем (включая Mac OS X) и Windows. Остановимся на этом подробнее. Важное отличие Windows от Linux (с точки зрения разработчика) заключается в отсутствии единого стандарта сборочных файлов. *Microsoft Visual Studio* использует свои файлы проектов, *C++ Builder* – свои, *MinGW* – свои. Преимущество *CMake* в том, что эта система способна генерировать «родные» сборочные файлы для всех перечисленных средств разработки (как и для многих других). Кроме того, CMake стремится максимально использовать фирменные средства генерации сборочных файлов – например, для проектов *Qt* применяется *qmake*. Вдобавок CMake обладает интеллектуальной системой поиска инструментов сборки и библиотек на конкретной платформе (интроспекция) и автоматического конфигурирования. Благодаря этому она сама устанавливает многие параметры сборочных файлов, которые в других случаях приходится задавать вручную. Например, в ОС Linux CMake найдет директорию, где находится требуемый программе набор виджетов, а под Windows вам не придется указывать ей, где установлены *Visual Studio* или *C++ Builder*. Все это делает пакет CMake весьма простым и удобным в использовании.

Хотя в этих статьях мы будем рассматривать работу CMake в основном на примере программ, написанных на C++, она может пригодиться и тем, кто пишет на Java (и даже Fortran).

Схематично работу CMake можно описать следующим образом: вначале создается файл **CMakeLists.txt**, в котором описываются параметры сборки (расположение исходных текстов, требуемые внешние модули, цели сборки). Далее этот файл передается утилите *cmake*. Результатом ее работы является файл, содержащий инструкции сборки приложения для конкретной платформы (make-файл GNU make, файл проекта *Visual Studio* и т.д.). Суть идеи заключается в том, что описание процесса сборки в файле **CMakeLists.txt** абстрагировано как от конкретных особенностей отдельных систем (расположение файлов, возможности компиляторов), так и целых платформ. Читая общее описание процесса сборки из файла **CMakeLists.txt**, программа *cmake* создает файл инструкций сборки, учитывающий специфику конкретной системы.

Как же CMake удается быть такой «умной»? В ее основе лежит мощный скриптовый язык, который используется как в файлах **CMakeLists.txt**, так и в специальных сценариях – модулях. Именно в модулях реализованы такие функции CMake, как поиск файлов и проверка возможностей системы. Файлы модулей расположены в поддиректории **Modules** каталога **cmake**. Их можно разделить на три категории. Одна группа модулей предназначена для настройки работы системы на конкретной платформе (Linux, Winodws, Mac OS, BeOS и т.д.), вторая обеспечивает интеллектуальный поиск средств разработки и различных вспомогательных утилит, а также проверку их возможностей, а третья разыскивает библиотеки, требуемые отдельным программам, и определяет их параметры.

Важную роль в работе CMake играют также файлы шаблонов для генерации различных файлов сборки. Поскольку файлы модулей и шаблонов отделены от самой утилиты *cmake* и написаны на том же языке сценариев, что и файлы описания сборки, расширить возможности CMake очень просто. CMake 2.6 поставляется с тремя сотнями модулей, которые адаптируют пакет для работы с множеством разных инструментов, начиная с компиляторов C++ и Fortran и интерпретаторов Perl и Python и заканчивая такими библиотеками, как *OpenSceneGraph*, *SDL*, *Qt 4* и *wxWidgets*. Файлы модулей поставляются не только с дистрибутивами CMake, но и вместе с некоторыми библиотеками, разработчики которых уже заинтересованы в поддержке CMake.

В путь

Хотя основные преимущества CMake связаны с кросс-платформенностью, что и отражено в названии пакета, простота работы с различными библиотеками может сделать CMale полезным и для Linux-проектов, которые не предполагают переноса на другие платформы. Благодаря CMake вы получаете простой и стандартный способ сборки приложений, использующих *Qt*, *KDE*, *GTK+*, *wxWidgets*...

Приступая к работе

Все примеры, рассмотренные в этих статьях, ориентированы на CMake 2.6 и выше (актуальная на момент написания статьи версия – 2.6.1). Если у вас установлена более ранняя версия CMake, обновите ее – это несложно (скачать дистрибутив можно с сайта www.cmake.org, он также есть и на **LXF**DVD). Хотя изложенный здесь материал ориентирован в основном на Linux-программиста, мы не будем пренебрегать кросс-платформенностью CMake, так что если вы все еще пользуетесь ОС из Редмонда, рекомендуется установить CMake и на нее тоже. Для изучения всех возможностей CMake мы воспользуемся кросс-платформенным набором виджетов *wxWidgets* (последнюю версию оного я тоже рекомендую установить на все интересующие вас платформы с сайта wxwidgets.org, см. также **LXF98-99**).

Если вы никогда не программировали с *wxWidgets* и не намерены делать этого впредь, могу вас успокоить – мы и не будем заниматься программированием с *wxWidgets*. Мы изучаем систему сборки программ, а не их написания, поэтому в качестве учебного материала нам вполне подойдут многочисленные примеры, входящие в состав дистрибутива *wxWidgets*. Поскольку мы практически не будем касаться их внутренней начинки, почти все наши рекомендации и инструкции можно рассматривать как универсальные, применимые при работе с любой другой библиотекой виджетов. Нашим основным средством разработки будет, естественно, GCC, а в обзоре переноса проектов под Windows мы рассмотрим (очень бегло) сборку проектов с помощью *Visual Studio* и *C++ Builder*.

Первый пример

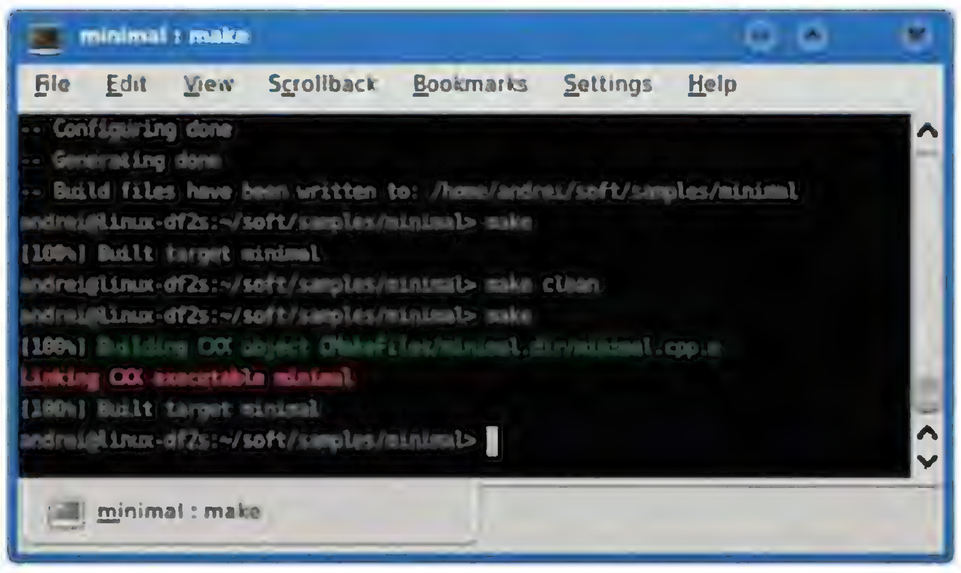
В директории **samples** дистрибутива *wxWidgets* найдите поддиректорию **minimal**. Она содержит минимальный пример приложения *wxWidgets*. Здесь также находятся файлы сборки приложения на все случаи жизни; а вот **CMakeLists.txt** среди них нет. Давайте исправим это упущение! Ниже приводится исходный текст файла **CMakeLists.txt** (который вы найдете на диске) для приложения *minimal*. Чтобы упростить работу, мы разместим его в той же директории, что и исходные тексты приложения.

```
project(minimal)
cmake_minimum_required(VERSION 2.6)
set(wxWidgets_USE_LIBS base core)
find_package(wxWidgets REQUIRED)
include(${wxWidgets_USE_FILE})
set(minimal_SRCS minimal.cpp)
if(WIN32)
set(minimal_SRCS ${minimal_SRCS} minimal.rc)
endif(WIN32)
add_executable(minimal WIN32 ${minimal_SRCS})
target_link_libraries(minimal ${wxWidgets_LIBRARIES})
```

Те, кому сразу не терпится попробовать, могут скомандовать в директории **minimal**

```
cmake ./
```

после чего вызывать *make*. Процесс сборки с помощью make-файла, сгенерированного CMake, выглядит довольно красочно (**рис. 1**).



› Рис. 1. Make-файлы CMake в консоли Linux.

После окончания процесса можно скомандовать

```
minimal
```

и посмотреть на работу программы, которую мы не написали, но собрали (**рис. 2**).

› Рис. 2. Приложение wxWidgets под Linux.

Прежде чем разбирать этот пример, рассмотрим кратко структуру языка сценариев CMake. На протяжении всех последующих статей мы, в основном, будем заниматься его изучением. Элементы языка сценариев CMake можно разделить на три категории: переменные, свойства и команды. Переменные играют в проектах CMake такую же важную роль, как и в файлах make. Настройка параметров сборки проекта выполняется, как правило, путем присвоения им различных значений. С помощью свойств проект CMake получает сведения о состоянии системы, для которой генерируются сборочные файлы. Свойства можно рассматривать как переменные, доступные только для чтения. Значения свойств устанавливаются средой CMake. Команды, как вы, конечно, догадались, выполняют различные действия с переменными и свойствами CMake и управляют процессом генерации сборочных файлов. Важной особенностью языка CMake является его расширяемость. Модули CMake, о которых говорилось выше, позволяют определять новые перемен-

Где найти документацию

Общее описание синтаксиса языка CMake можно найти в Интернете на странице cmake.org/HTML/Documentation.html. Там же есть ссылка на бумажную книгу Кена Мартина [Ken Martin] и Билла Хоффмана [Bill Hoffman], «Mastering CMake» (актуальное издание – четвертое, на русском языке не выходило). Что касается расширений CMake, описания многих из них входят в стандартную документацию. Описания остальных расширений следует искать на сайтах тех проектов, для которых они предназначены.

Не следует думать, что расширения языка CMake чрезмерно его усложняют. Базовый набор команд переменных и свойств языка не очень велик, а расширения, как правило, связаны с конкретными пакетами, средами и платформами разработки. Если вы не ставите перед собой цель стать эрудитом в области CMake, вам потребуется знать только те расширения, которые нужны для используемых вами инструментов, а их, как правило, тоже не много.

Интересно отметить, что очень подробная справка по командам CMake включена прямо в исполняемый файл утилиты: наберите **cmake --help-full | less**, и вы поймете, что мы имеем в виду.

ные, свойства и команды. Именно благодаря расширяемости языка CMake этот пакет может использоваться для сборки столь разных проектов.

Перейдем теперь к рассмотрению файла **CMakeLists.txt**. Описание сборки начинается с команды project().

С большой буквы или с маленькой?

Язык CMake обладает частичной регистронезависимостью. Имена команд нечувствительны к регистру (IF(), If(), if() и iF() – одно и то же), в то время как имена свойств и переменных – чувствительны. В связи с регистронезависимостью имен команд возникает вопрос, как правильнее писать – с большой буквы, с маленькой или полностью заглавными буквами. Разные авторы пишут по-разному (в документации CMake имена команд пишутся полностью строчными буквами, а во многих файлах CMake, написанных самими разработчиками – заглавными). Лично я пишу имена команд с маленькой буквы, и буду следовать этому принципу в статьях данной серии.

Команда project() задает имя проекта, а также позволяет указать (в виде необязательного параметра) язык программирования. В качестве имени проекта мы взяли название компилируемой программы (это не обязательное правило CMake, но так удобнее). Язык программирования (C, CXX, Java) можно опустить. В этом случае CMake попробует «угадать» его по расширениям файлов исходных текстов. Команда project() – не просто элемент оформления файла **CMakeLists.txt**. Помимо прочего, она определяет переменные projectname_SOURCE_DIR и projectname_BINARY_DIR. Как нетрудно догадаться, последние содержат имена директории исходных тестов (обычно тот же каталог, где расположен файл **CMakeLists.txt**) и директории, в которой будет сохранен результат сборки. Файлы программ по умолчанию собираются в той же директории, где расположен файл **CMakeLists.txt**, а файлы библиотек – в поддиректории **lib** этого каталога.

Команда cmake_minimum_required() позволяет задать минимальный номер версии CMake, требуемый для сборки проекта. Она не является обязательной, но весьма полезна, если файл **CMakeLists.txt** использует возможности, появившиеся в последней версии CMake, которая пока что есть не у всех. Если установленная в системе версия CMake не соответствует минимальным требованиям файла **CMakeLists.txt**, будет выдано сообщение об ошибке.

Следующая команда, set(), является одной из наиболее часто используемых. Она позволяет присваивать переменным значения. Первым аргументом команды должно быть имя переменной (если переменной с таким именем ранее не существовало, она будет создана), далее следует список значений, которые ей присваиваются (одной переменной можно присвоить сразу несколько значений). Например, в команде

```
set(wxWidgets_USE_LIBS base core)
```

мы присваиваем переменной wxWidgets_USE_LIBS (которая должна содержать список требуемых программе модулей wxWidgets) значения base и core. Переменные CMake можно использовать рекурсивно, то есть добавлять новые значения к списку уже имеющихся, как например, в строке

```
set(minimal_SRCS ${minimal_SRCS} minimal.rc)
```

где к значениям переменной minimal_SRCS добавляется еще одно. Обратите внимание на конструкцию ${имя_переменной}. Она возвращает значение переменной с указанным именем (без символа $ и фигурных скобок CMake «не понял» бы, что тут имеется в виду имя переменной). Правила использования имен и значений переменных в CMake похожи на правила для переменных окружения в сценариях оболочки.

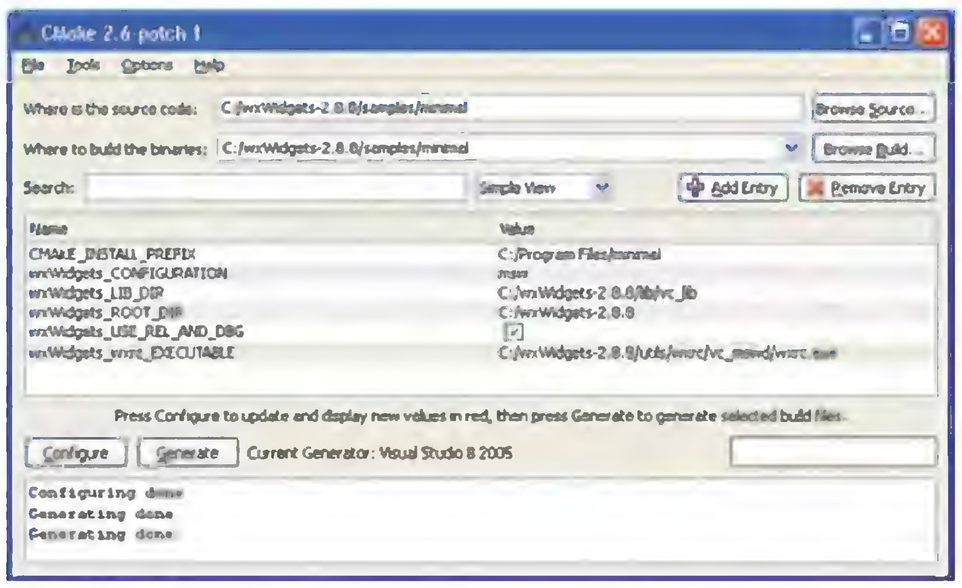
Команда find_package() – одна из важнейших в CMake. В следующей статье мы остановимся на ней подробнее; сейчас скажем только, что find_package() загружает расширения CMake. В нашем примере загружается расширение *wxWidgets*, предназначенное для генерации файлов сборки программ, использующих *wxWidgets*. Важно подчеркнуть, что команда find_package() не ищет сами библиотеки, она только загружает расширение CMake. Поиск библиотек, определение их версий и прочие действия, связанные с библиотеками, выполняет загруженное расширение. Естественно, что поскольку система CMake сама не выполняет сборку приложений, библиотеками она «интересуется» лишь с точки зрения генерации инструкций для сборочных файлов. В нашем примере команда find_package() вызывается с опцией REQUIRED. Тем самым мы указываем CMake, что если расширение для работы с wxWidgets не найдено, генерация файлов, необходимых для сборки приложения, выполняться не может.

Команда add_executable() относится к числу тех, что устанавливают цели сборки. Собственно, она указывает, что ею является исполняемый файл программы. Если бы мы собирали библиотеку, следовало бы использовать команду add_library(). Команда add_executable() позволяет указать имя собираемого файла и связать с ним исходные тексты. Обратите внимание на опцию WIN32, переданную команде add_executable(). Ее потребовалось упомянуть для того, чтобы на платформе Windows создавались файлы сборки графического приложения (а не консольного, как это будет по умолчанию). В Linux нет принципиальной разницы между структурой исполняемого файла графической и консольной программы (в Windows – есть, интересующихся читателей отсылаю к соответствующей литературе). Как же отреагирует CMake для Linux на присутствие опции WIN32? Правильный ответ – никак, пакет ее просто проигнорирует. Помимо файлов программ и библиотек, в проектах CMake можно задавать и другие цели, например, установку и упаковку приложений.

Последняя команда файла, target_link_libraries(), указывает CMake, какие инструкции следует добавить в сборочный файл для подключения к проекту сторонних библиотек. Первый ее аргумент – цель, определенная выше командой add_executable() (вообще говоря, целей сборки может быть несколько), далее идет список внешних библиотек, в которых она нуждается. В нашем примере он хранится в переменной wxWidgets_LIBRARIES, определенной в расширении wxWidgets. На содержимое переменной wxWidgets_LIBRARIES влияет переменная wxWidgets_USE_LIBS, вот почему значение ей следует присвоить до обращения к команде find_package().

А что в Windows?

Наш файл **CMakeLists.txt** уже содержит некоторые элементы, которые пригодятся при генерации сборочных файлов под Windows. Тем не менее, на этой платформе нас ждут некоторые сложности. Вызваны они тем, что файловая система в Windows гораздо более хаотична, чем в Linux. В Unix-системах существуют устойчивые традиции касательно того, где должны быть расположены те или иные файлы. Например, файлы разделяемых библиотек следует искать (и устанавливать) в директориях **/lib**, **/usr/lib**, **/usr/local/lib**. В Windows путаницы гораздо больше. Файлы тех же разделяемых библиотек могут располагаться либо в системных каталогах Windows, либо в собственных каталогах программ. В дополнение к этому файловая система Windows не имеет общего корня, как это принято в Unix-системах. Все эти особенности приводят к тому, что под Windows пакету CMake гораздо труднее найти места расположения требуемых библиотек. Принимая во внимание этот факт, а также то, что консоль в Windows гораздо менее популярна, разработчики CMake для Windows добавили в пакет специальную утилиту с графическим интерфейсом (**рис. 3**).



› Рис. 3. CMake GUI для Windows.

Главная задача CMake GUI – генерация файлов сборки приложения для различных сред, используемых на платформе Windows. Попутно эта утилита позволяет присвоить значения некоторым переменным CMake, которые сам пакет определить не смог (или «не уверен» в их правильности). Отметим, что эти дополнительные настройки влияют на получаемые в результате генерации файлы сборки, но не влияют на содержимое самого файла **CMakeLists.txt**.

Создание файлов сборки с помощью CMake GUI состоит из двух этапов: настройки и собственно генерации. Первая операция выполняется по команде Configure, вторая – по команде Generate. В процессе конфигурации утилита позволяет вам выбрать среду разработки из довольно внушительного списка и настроить значения «сомнительных» переменных. В процессе генерации утилита может выявить синтаксические ошибки в файле **CMakeLists.txt**. Если в качестве целевой среды сборки выбрана *Visual Studio*, CMake для Windows генерирует файл **ALL_BUILD.vcproj**. Если же целью является компилятор Borland C++, в результате создается make-файл для консольной версии компилятора (**рис. 4**).

› Рис. 4. Консольная сборка под Windows выглядит почти так же, как и под Linux. Вот она, кросс-платформенность!

Выполнив сборку, мы получаем то же приложение, что и под Linux.

В этой статье мы лишь бегло ознакомились с языком CMake. В следующий раз нас ждет более подробное знакомство с его переменными и командами, а также с ключами, позволяющими управлять консольным вариантом программы. LXF

» Через месяц Мы научимся говорить на языке CMake более-менее бегло.

Работа с двумя переменными

Повествование об R возобновляется на новом уровне. С января по апрель этого года мы провели предварительный обзор возможностей, а теперь **Антону Коробейникову** и **Евгению Балдину** предстоит настоящая работа.

Наши эксперты

Антон Коробейников
Молодой преподаватель, читающий спецкурсы по применению R для решения различных статистических задач студентам кафедры статистического моделирования матмеха СПбГУ.

Евгений Балдин
Физик, преподаватель, научный сотрудник ИЯФ им. Будкера, профессионально занимается обработкой экспериментальных данных в области физики высоких энергий. Давний сторонник свободного ПО, впечатлённый мощью R.

Эмблема R взята с официального сайта проекта http://developer.r-project.org/Logo/

ЧАСТЬ 1 Проверка гипотез однородности

Две разные выборки называются однородными, если они одинаково распределены. Проверка «гипотез однородности» в математической статистике занимает особое место. Этот факт связан с тем, что для практических приложений, как правило, характерны задачи сравнения двух и более групп наблюдений. Сравнивать выборки можно совершенно разными способами: исследователя может интересовать различие в средних, медианах, дисперсиях при разнообразных предположениях относительно самих наблюдений.

Параметрические критерии проверки однородности выборок

Величины, имеющие нормальное распределение, возникают в реальных экспериментах совершенно естественным образом. При измерении любой характеристики всегда имеется ошибка измерения. Если предположить, что погрешность прибора имеет нормальное распределение, то среднее отвечает за систематическую ошибку, а дисперсия – за величину случайной ошибки. Поэтому критерии, представленные в данном разделе учебника, предполагают, что выборка имеет нормальное распределение. Если это заранее не известно, то данный факт нужно проверять отдельно (см. *Проверка гипотез нормальности распределения* ниже). В противном случае все выводы, полученные на основе этих критериев, будут ошибочными.

Двухвыборочный критерий Стьюдента равенства средних

Двухвыборочный t-критерий используется для проверки гипотезы о равенстве средних в двух независимых выборках, имеющих нормальное распределение. В своей классической постановке критерий приводится в предположении равенства дисперсий. Однако в **R** реализована его модификация, позволяющая избавиться от данного предположения.

Воспользуемся классическим набором данных, который фигурировал ещё в оригинальной статье Стьюдента (псевдоним Вильяма Сили Госсета [William Sealy Gosset]). В упомянутой работе производилось сравнение влияния двух различных снотворных на увеличение продолжительности сна (см. **рис. 1**). В **R** этот массив данных доступен под названием sleep в пакете stats. В столбце extra содержится среднее приращение продолжительности сна после начала приёма лекарства (по отношению к контрольной группе), а в столбце group – код лекарства (первое или второе).

```
> plot(extra ~ group, data = sleep)
```

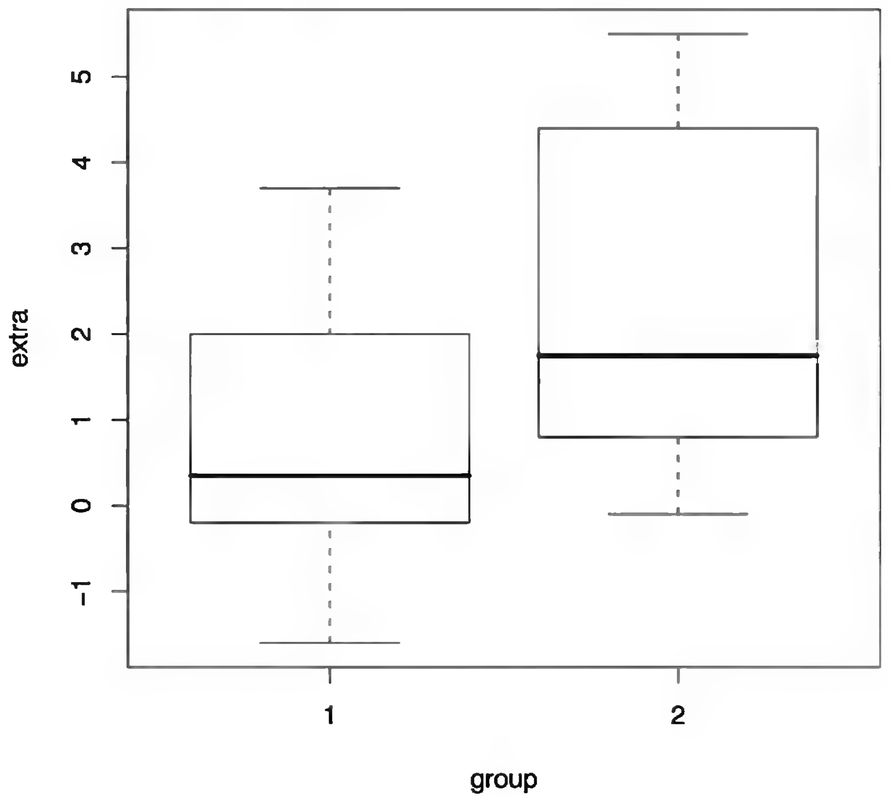


Рис. 1. Среднее приращение продолжительности сна после начала приёма разных лекарств в двух группах, по отношению к контрольной.

Влияние лекарства на каждого человека индивидуально, но среднее увеличение продолжительности сна можно считать вполне логичным показателем «силы» препарата. Основываясь на этом предположении, попробуем проверить при помощи t-критерия, значимо ли различие в средних для двух этих выборок (соответствующих двум разным лекарствам):

```
> with(sleep, t.test(extra[group == 1],
+            extra[group == 2], var.equal = FALSE))
            Welch Two Sample t-test
data:  extra[group == 1] and extra[group == 2]
t = -1.8608, df = 17.776, p-value = 0.0794
```

```
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -3.3654832  0.2054832
sample estimates:
mean of x mean of y
     0.75      2.33
```

Параметр var.equal позволяет выбрать желаемый вариант критерия: оригинальный t-критерий Стьюдента в предположении одинаковых дисперсий (TRUE) или же t-критерий в модификации Уэлча [Welch], свободный от данного предположения (FALSE).

Хотя формально гипотеза о равенстве средних не отвергается на стандартных уровнях значимости, мы видим, что возвращаемое p-значение (0.0794) достаточно маленькое, поэтому к данном результату следует отнестись критически. Это означает, что, возможно, стоит попробовать другие методы для проверки гипотезы, увеличить количество наблюдений, ещё раз убедиться в нормальности распределений, и т.д.

Можно ли использовать t-критерий, если необходимо сравнить среднее в *зависимых* выборках (например, при сравнении какого-либо жизненного показателя у пациента до и после лечения)? Ответ на этот вопрос: «Да, можно, но не обычный, а модифицированный специальным образом под такую процедуру». В литературе это известно как *парный t-критерий*. Для использования парного t-критерия в **R** достаточно выставить опцию paired в TRUE:

```
> with(sleep, t.test(extra[group == 1],
+            extra[group == 2], paired = TRUE))
            Paired t-test
data:  extra[group == 1] and extra[group == 2]
t = -4.0621, df = 9, p-value = 0.002833
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -2.4598858 -0.7001142
sample estimates:
mean of the differences
                  -1.58
```

Здесь видно, что парный t-критерий отвергает гипотезу о равенстве средних с достаточно большой надёжностью. Следует отметить, что использованные в этом примере выборки не были «парными», поэтому, строго говоря, применять парный t-критерий нельзя, так что все выводы носят сугубо иллюстративный характер.

» Двухвыборочный критерий Фишера равенства дисперсий

Естественной характеристикой «размаха» распределения при нормальной модели является дисперсия. Предположим, что нам потребовалось проверить гипотезу об отсутствии различий в дисперсиях двух выборок, и при этом не хочется делать абсолютно никаких допущений относительно значений средних. Такая задача может возникнуть, например, при сравнении точности двух приборов. Систематическую ошибку иногда можно уменьшить за счёт лучшей калибровки, случайную же погрешность убрать почти никогда нельзя. В связи с этим задача проверки равенства дисперсий (например, при сравнении эталонного и поверяемого приборов) становится достаточно актуальной.

Решением такой задачи служит F-критерий Фишера [Fisher]. В **R** он реализован в функции var.test():

```
> x <- rnorm(50, mean = 0, sd = 2);
> y <- rnorm(30, mean = 1, sd = 1);
> var.test(x, y)
            F test to compare two variances
data:  x and y
F = 3.8414, num df = 49, denom df = 29, p-value = 0.0002308
alternative hypothesis: true ratio of variances is not equal to 1
95 percent confidence interval:
 1.930003 7.227256
sample estimates:
ratio of variances
          3.841391
```

В данном примере участвуют две выборки с разным количеством наблюдений (50 и 30 для x и y, соответственно), разными средними (0 и 1) и дисперсиями (2^2=4 и 1^2=1). Гипотеза о равенстве дисперсий безусловно отвергается (p-значение мало). Кроме самого значения тестовой статистики, p-значения и величин степеней свободы, функция выводит оценку отношения дисперсий и доверительный интервал от него. Значение оценки 3.55 не очень сильно отличается от истинного значения отношения дисперсий 4.

На самом деле, критерий проверяет несколько более общую гипотезу, а именно, что отношение дисперсий двух выборок имеет какое-то наперёд заданное значение. Очевидно, равенство дисперсий является её частным случаем.

Предполагаемое значение отношения дисперсий можно задать с помощью опции ratio:

```
> x <- rnorm(50, mean = 0, sd = 2);
> y <- rnorm(30, mean = 1, sd = 1);
> var.test(x, y, ratio = 4)
            F test to compare two variances
data:  x and y
F = 1.1097, num df = 49, denom df = 29, p-value = 0.7778
alternative hypothesis: true ratio of variances is not equal to 4
95 percent confidence interval:
 2.230136 8.351157
sample estimates:
ratio of variances
          4.43876
```

Здесь видно, что при задании истинного значения отношения дисперсий гипотеза не отвергается.

Непараметрические критерии проверки однородности выборок

Критерии, приведённые в предыдущем разделе, работают только в предположении нормальности распределения данных. А что делать, если заранее известно, что выборки имеют другое распределение, или по каким-либо причинам проверить нормальность не получается? В таких случаях используются так называемые *непараметрические* критерии, т.е. критерии, свободные от предположения какой-либо параметрической модели данных. Естественно, ввиду того, что они оперируют гораздо меньшим «объёмом информации», эти критерии не смогут заметить те тонкие различия, которые были бы обнаружены при использовании их параметрических аналогов.

» Критерий Вилкоксона

Критерий Вилкоксона [Wilcoxon], двухвыборочный вариант которого ещё известен под именем критерия Манна–Уитни, является непараметрическим аналогом t-критерия.

Стандартный набор данных airquality содержит информацию о количестве озона в воздухе города Нью-Йорка с мая по сентябрь 1973 года. Проверим, например, гипотезу о том, что распределения уровня озона в мае и в августе было одинаковым:

```
> wilcox.test(Ozone ~ Month, data = airquality,
+             subset = Month %in% c(5, 8))
        Wilcoxon rank sum test with continuity correction
data:  Ozone by Month
W = 127.5, p-value = 0.0001208
alternative hypothesis: true location shift is not equal to 0
```

Критерий отвергает гипотезу о согласии распределений уровня озона в мае и в августе с достаточно большой надёжностью. В принципе, это достаточно легко вытекает из «общих соображений», т.к. как уровень озона в воздухе сильно зависит от солнечной активности, температуры и ветра. И если с солнечной активностью всё в порядке:

```
> wilcox.test(Solar.R ~ Month, data = airquality,
+             subset = Month %in% c(5, 8))
        Wilcoxon rank sum test with continuity correction
data:  Solar.R by Month
W = 422.5, p-value = 0.4588
```

```
alternative hypothesis: true location shift is not equal to 0
```

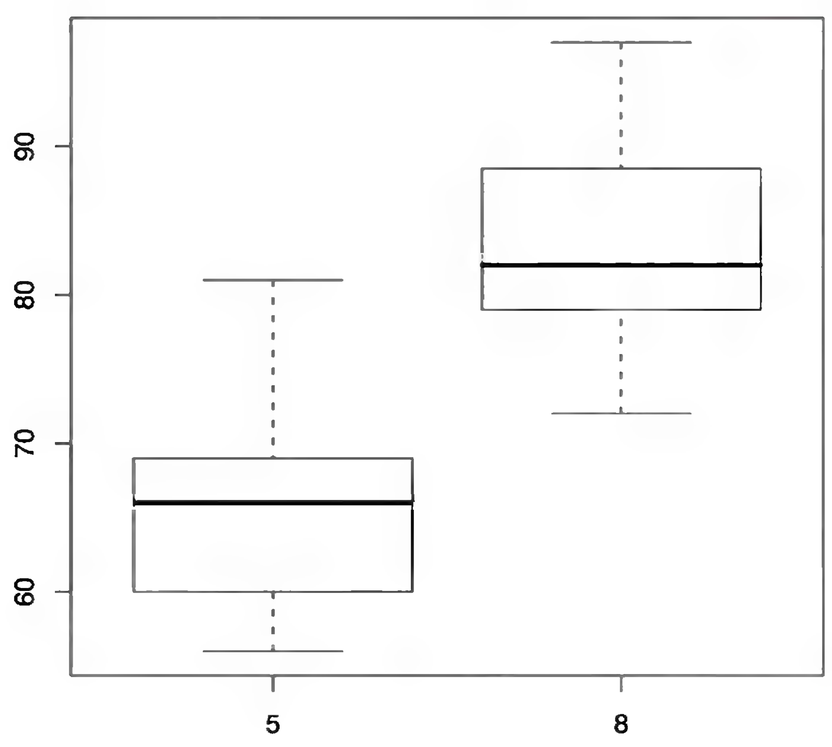
то распределения ветра и температуры, напротив, сильно различаются:

```
> wilcox.test(Temp ~ Month, data = airquality,
+                     subset = Month %in% c(5, 8))
        Wilcoxon rank sum test with continuity correction
data:  Temp by Month
W = 27, p-value = 1.747e-10
alternative hypothesis: true location shift is not equal to 0
> wilcox.test(Wind ~ Month, data = airquality,
+                     subset = Month %in% c(5, 8))
        Wilcoxon rank sum test with continuity correction
data:  Wind by Month
W = 687.5, p-value = 0.003574
alternative hypothesis: true location shift is not equal to 0
```

Эти различия хорошо видны, скажем, на изображении «ящиков с усами» (**рис.2**):

```
> boxplot(Temp ~ Month, data = airquality,
+                     subset = Month %in% c(5, 8))
```

но в более сложных случаях объективные результаты можно получить только с использованием критериев.



› Рис. 2. Распределение температуры в Нью-Йорке в мае и в августе 1973 года.

По умолчанию, критерий проверяет гипотезу о том, что распределения двух выборок отличаются лишь постоянным и известным сдвигом (который, в свою очередь, по умолчанию равен нулю). Задать его можно при помощи параметра mu. Например:

```
> x <- rnorm(50, mean = 0);
> y <- rnorm(50, mean = 2);
> wilcox.test(x, y);
        Wilcoxon rank sum test with continuity correction
data:  x and y
W = 230, p-value = 2.091e-12
alternative hypothesis: true location shift is not equal to 0
> wilcox.test(x, y, mu = -2);
        Wilcoxon rank sum test with continuity correction
data:  x and y
W = 1335, p-value = 0.5602
alternative hypothesis: true location shift is not equal to -2
```

Критерий Манна–Уитни, как и t-критерий, бывает парным. Для использования парной модификации необходимо выставить опцию paired в значение TRUE. Задание опции conf.int позволяет получить оценку различия в сдвиге между двумя выборками (естественно, при условии, что кроме сдвига, распределения двух выборок ничем не отличаются) и доверительный интервал для него:

```
> x <- rnorm(50, mean = 0);
> y <- rnorm(50, mean = 2);
> wilcox.test(x, y, conf.int = TRUE);
        Wilcoxon rank sum test with continuity correction
data:  x and y
W = 227, p-value = 1.803e-12
alternative hypothesis: true location shift is not equal to 0
95 percent confidence interval:
 -2.418617 -1.500210
sample estimates:
difference in location
            -1.941988
> wilcox.test(x, y, mu = -2);
        Wilcoxon rank sum test with continuity correction
data:  x and y
W = 1292, p-value = 0.7748
alternative hypothesis: true location shift is not equal to -2
```

Гипотеза о полном равенстве распределений была отвергнута, а гипотеза о том, что одно распределение отличает от другого просто сдвигом – принята, что и требовалось доказать.

» Непараметрические критерии сравнения масштаба

Дисперсия является естественным параметром масштаба для выборки из нормальной совокупности. Этот факт в случае нормального распределения и позволяет заменить гипотезу о совпадении масштабов на гипотезу о совпадении дисперсий. При отказе от нормальной модели дисперсия уже не является характеристикой масштаба, и поэтому надо честно проверять гипотезу именно о совпадении масштабов распределений двух выборок.

Формально предполагается, что одна из выборок имеет распределение с плотностью **f(x-a)**, другая – **s*f(s*(x-a))**. Здесь функция плотности **f** и параметр сдвига **a** считаются неизвестными. Мы заинтересованы в проверке совпадения масштабов у двух выборок, т.е. того, что **s=1**.

В стандартном пакете stats реализованы два классических непараметрических критерия, позволяющих проверить равенство масштабов: критерий Ансари–Брэдли [Ansari–Bradley] и критерий Муда [Mood]. Начнём с первого.

```
> ansari.test(runif(50), rucunif(50, max = 2))
        Ansari-Bradley test
data:  runif(50) and runif(50, max = 2)
AB = 1404, p-value = 0.07526
alternative hypothesis: true ratio of scales is not equal to 1
```

Распределение выборок в данном случае отличается только масштабом. Критерий отвергает гипотезу о совпадении дисперсий на стандартных уровнях значимости, как это и должно быть. Для критерия Муда всё аналогично:

```
> mood.test(runif(50), runif(50, max = 2))
        Mood two-sample test of scale
data:  runif(50) and runif(50, max = 2)
Z = -2.5685, p-value = 0.01021
alternative hypothesis: two.sided
```

ЧАСТЬ 2 Проверка гипотез нормальности распределения

Большая часть инструментов статистического вывода работает в предположении о том, что выборка получена из нормальной совокупности. За примерами далеко ходить не надо: t-критерий, критерий Фишера, построение доверительных интервалов для линейной регрессии и проверка гипотезы о линейной независимости двух выборок.

Как уже отмечалось ранее, нормальное распределение естественным образом возникает практически везде, где речь идёт об измерении с ошибками. Более того, в силу центральной предельной теоремы, распределение многих выборочных величин (например, выборочного среднего) при достаточно больших объёмах выборки хорошо аппроксимируется нормальным распределением, вне зависимости от исходного распределения.

В связи с этим становится понятным, почему проверке распределения на нормальность стоит уделить особое внимание. В дальнейшем речь пойдёт о так называемых *критериях согласия* (goodness-of-fit tests). Проверяться будет не просто факт согласия с нормальным распределением с определёнными фиксированными значениями параметров, а несколько более общий факт принадлежности распределения к семейству нормальных со всевозможными значениями параметров.

Основные классические критерии проверки на нормальность собраны в пакете nortest. Его можно установить со CRAN при помощи вызова функции install.packages():

```
> install.packages(pkgs=c(«nortest»))
```

При этом будет предложено выбрать зеркало для загрузки файла Подключить установленный пакет можно при помощи функции library():

```
> library(nortest)
```

Может возникнуть вопрос: «А зачем так много разных критериев для проверки одного факта? Нельзя ли выбрать наилучший и всегда его использовать?». Ответ неутешителен: «В общем случае, к сожалению, нельзя». Попробуем это объяснить.

Напомним, что ошибкой первого рода статистического критерия называется факт принятия альтернативной гипотезы в ситуации, когда верна гипотеза по умолчанию. Например, пусть статистический критерий используется для разграничения доступа к какому-нибудь ресурсу. Тогда отказ в доступе для авторизованного пользователя и будет ошибкой первого рода для такого критерия. Ясно, что возможна «симметричная» ошибочная ситуация, заключающаяся в предоставлении доступа к ресурсу неавторизованному пользователю. Такая ошибка называется *ошибкой второго рода*: принятие критерием гипотезы по умолчанию в ситуации, когда она неверна (т.е. имеет место альтернативная гипотеза).

Как правило, чувствительность критерия к ошибкам первого рода мы настраиваем самостоятельно (как раз выбором тех самых «стандартных уровней значимости», с которым и сравниваем выдаваемое p-значение). С ошибкой второго рода всё гораздо хуже: её вероятность сильно зависит от выбранной альтернативной гипотезы и является неотъемлемой характеристикой самого критерия. В редких случаях удаётся построить критерий, который является наилучшим (это так называемые *равномерно наиболее мощные критерии*), и, к сожалению, это невозможно для нашей задачи. В данном случае нулевая гипотеза формулируется просто: *выборка имеет нормальное распределение с некоторыми неизвестными параметрами*, а альтернативная гипотеза – это полное её отрицание. Альтернативная гипотеза гораздо «богаче» нулевой: в нее входят все распределения, отличные от нормального. Для того и понадобилась целая батарея критериев: одни работают лучше против одного семейства альтернатив, вторые – против другого. Использование всего набора позволяет быть хоть как-то уверенным в том, что распределение, не являющееся нормальным, будет «распознано» хотя бы одним из критериев.

Критерий Лиллифорса

Критерий Лиллифорса [Lilliefors] является вариантом известного классического критерия Колмогорова–Смирнова, специально модифицированного для проверки нормальности. Эта модификация существенна. Для проверки гипотезы нормальности *нельзя* использовать классический непараметрический критерий Колмогорова–Смирнова, реализованный в функции ks.test(). Критерий Лиллифорса реализован в функции lillie.test():

```
> lillie.test(rnorm(100, mean = 6, sd = 4));
        Lilliefors (Kolmogorov-Smirnov) normality test
data:  rnorm(100, mean = 6, sd = 4)
D = 0.0463, p-value = 0.8621
> lillie.test(runif(100, min = 2, max = 4));
        Lilliefors (Kolmogorov-Smirnov) normality test
data:  runif(100, min = 2, max = 4)
D = 0.0732, p-value = 0.2089
```

Критерии Крамера – фон Мизеса и Андерсона–Дарлинга

Первый критерий известен в русскоязычной литературе под именем критерия ω^2 или критерия Смирнова. Эти критерии менее известны, но обычно работают гораздо лучше, нежели критерий Лиллифорса. Они реализованы в функциях cvm.test() и ad.test() соответственно:

```
> cvm.test(rnorm(50, mean = 6, sd = 4));
        Cramer-von Mises normality test
data:  rnorm(50, mean = 6, sd = 4)
W = 0.0321, p-value = 0.8123
> ad.test(runif(50, min = 2, max = 4));
        Anderson-Darling normality test
data:  runif(50, min = 2, max = 4)
A = 1.5753, p-value = 0.0004118
```

Критерий Шапиро–Франсиа

Данный критерий работает достаточно хорошо в большинстве не очень «сложных» случаев. Получить p-значение можно посредством функции sf.test():

```
> sf.test(rexp(50, rate = 2));
        Shapiro-Francia normality test
data:  rexp(50, rate = 2)
W = 0.7803, p-value = 2.033e-06
```

Критерий хи-квадрат Пирсона

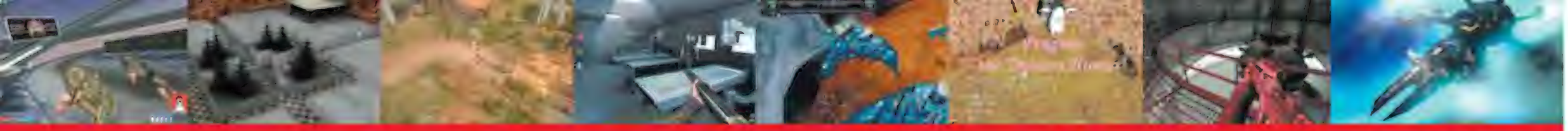
В отличие от задач проверки пропорций, критерий хи-квадрат обычно очень плохо работает в задачах проверки распределения на нормальность. Вероятность ошибки второго рода для достаточно широкого класса альтернативных распределений очень велика. В связи с этим, использовать его не рекомендуется.

Тем не менее, его реализация предоставлена функцией pearson.test(). У нее есть булева опция adjusted, которая позволяет внести поправки в p-значение из-за наличия двух неизвестных параметров. Рекомендуемая последовательность действий такая: получить два p-значения: одно – соответствующее adjusted=TRUE, второе – для adjusted=FALSE. Истинное p-значение обычно находится между ними. Кроме того, полезно поварьировать объем выборки и посмотреть, насколько сильно меняется p-значение. Если влияние объёма выборки сильное, то от использования критерия, во избежание ошибок, следует отказаться.

```
> pearson.test(rnorm(50, mean = 6, sd = 4));
        Pearson chi-square normality test
data:  rnorm(50, mean = 6, sd = 4)
P = 5.2, p-value = 0.6356
> pearson.test(runif(50, min = -1, max = 1));
        Pearson chi-square normality test
data:  runif(50, min = -1, max = 1)
P = 7.6, p-value = 0.3692
```

Теперь мы умеем не только отличать нормальное распределения от «ненормального», но и сравнивать их друг с другом. Это одна из самых первых ступенек на пути понимания сути данных, которые волей-неволей приходится собирать для познания природы абсолютно любых явлений. LXF

Через месяц Корреляции, таблицы сопряжённости и графическое представление данных.



LINUX FORMAT

» GLSL Проникаем в тайны видеоускорителя с программами на шейдерах.

АНДРЕЙ ПРАХОВ
Участник нескольких игровых проектов, представитель СМИ и начальник www.linuxmedia.ru

Хорошее продолжение

Когда-то, а точнее, в феврале текущего года, Blender Institute анонсировал новый открытый проект – *Apricot*. Для тех, кто не в курсе – это игра, основанная на сюжете мультфильма *Big Buck Bunny*, которая должна была продемонстрировать профессиональному миру пригодность *Blender*'а к созданию таких проектов. Для этой цели команда разработчиков решила заменить стандартный игровой движок BGE на небезызвестный, но не слишком распространенный Crystal Space.

Подобная акция с успехом была проведена еще в 2006 году, когда на суд общественности была вынесена мультипликационная короткометражка *Ele-phants Dream*, созданная, естественно, в *Blender*. Этим шагом разработчики доказали пригодность пакета к «серьезной» работе с 3D.

Что же касается нового амбициозного проекта, то и с ним все вроде бы «на мази». Совсем недавно в блогах разработчиков появилась информация о доступности к скачиванию демо первого уровня, где показаны основные фишки главного героя. Вот только одно «но» – на август месяц, согласно расписанию, планировалась уже финальная версия. Конечно, для Open Source задержка – явление нередкое, но так хочется увидеть, что же получилось в итоге, а главное – в полной мере насладиться новыми возможностями, которые обретет программа после столь масштабных изменений!

info@linuxmedia.ru

Нереальная реальность

ЧАСТЬ 1 Пластмасса, сталь, огонь, вода, облака, свет и процедурные текстуры – что объединяет все эти понятия? Оказывается, шейдеры – и **Андрей Прахов** сейчас расскажет о них подробнее.

Шейдер! Многочисленная армия поклонников игр так или иначе знакома с этим термином. Строго говоря, программы на шейдерах используются не только в играх, но и в системах 3D-моделирования, визуализации, спецэффектах. Для обычного игрока это слово вызывает разве что ассоциации с чем-то загадочным, непостижимым и, отчасти, крутым. Но мы же с вами не только игроки, а поэтому давайте вместе разбираться, что это за зверь.

Не секрет, что возможности трехмерных ускорителей (GPU) в своей области на порядок выше программных эквивалентов. Это и понятно, так как процессоры современных видеоплат представляют собой специализированные чипы, ориентированные на выполнение задач, связанных с просчетом выводимой графики и только (правда, в последнее время инженерами компании NVIDIA ведутся интенсивные исследования по привлечению мощностей GPU к вычислениям игровой [и неигровой, – *прим. ред.*] физики). Однако для задействования этих функций необходимо писать код, понятный конкретным GPU. Ранее для этого использовался низкоуровневый язык, чем-то похожий на ассемблер. К счастью, в настоящее время необходимость в нем отпала, и разработчики различных графических API предоставили пользователям свои версии высокоуровневых шейдерных языков. Так как в Linux применяется исключительно OpenGL, то и мы будем изучать созданный для него язык GLSL [OpenGL Shading Language]. Естественно, это не единственный язык подобного рода: так, для DirectX существует HLSL, компания NVIDIA предлагает свой вариант – Cg, а Pixar – RenderMan.

Прежде чем приступить к работе над шейдерами, необходимо оговорить ту программную среду, в которой мы будем ваять свои шедевры. После недолгого размышления я откинул идею об использовании «чистого» API OpenGL, так как тогда нам пришлось бы изучать не столько шейдеры, сколько саму библиотеку. Поэтому для тестирования и прогона наших программ мы возьмем систему моделирования *Blender*. В этом случае мы получаем ощутимые плюсы в виде легкости написания кода и мощной базы для эффективного отображения задуманных эффектов. Для любителей «чистого» OpenGL будут предусмотрены специальные сноски с необходимыми пояснениями. Обратите внимание! Эти уроки **только о GLSL**; информация о *Blender* и OpenGL будет в рамках, строго необходимых для работы (см. врезки).

Что может дать использование шейдеров в наших программах? Вот небольшой список для ознакомления:

» Реалистичность отображения материалов (дерево, металл, пластик, краски);

» Природные явления (огонь, вода, облака, дым);

» Процедурные текстуры;
» Способы обработки изображений (яркость, контрастность, сепии, сглаживание, искривления и т.п.);
» Анимация;
» Реалистичность эффектов освещения, преломления.

Перечень далеко не полный. Не забывайте о том, что для вычисления и демонстрации подобных эффектов с хорошей скоростью центральный процессор просто не годится. Благодаря шейдерам появляется возможность освободить его для более интересных задач, скажем, просчета AI.

Выделяют два вида шейдеров: вершинный и фрагментный. Каждый из них выполняет определенные операции на соответствующем процессоре. Запомните! При изменении стандартной программы обработки какого-либо шейдера необходимо выполнить весь «затертый» функционал в своей реализации кода. Например, невозможно одновременно использовать стандартные функции преобразования вершины и нормали и свой код для вычисления освещения. Написанный вами шейдер должен уметь выполнять все три перечисленные операции. К счастью, в GLSL существуют команды-эквиваленты для замещения стандартных действий. Для более полного понимания происходящего обратите внимание на **рис. 1**, где схематично представлен конвейер обработки данных OpenGL.

› Рис. 1. Стандартный конвейер операций OpenGL с программируемыми процессорами шейдеров.

» **Вершинный шейдер [vertex shader]** – это программа, выполняемая графическим процессором и обрабатывающая данные вершин конкретного примитива. Заметьте, что vertex shader работает исключительно с одной из вершин, не подозревая о существовании других. Таким образом, вы не можете добавлять или удалять вершины из существующего примитива. Для ускорения обработки современные GPU имеют до нескольких сотен соответствующих логических блоков.

Список возможных операций вершинного шейдера:

» Преобразование вершин и нормалей;
» Генерирование и преобразование текстурных координат;
» Настройки освещения;
» Взаимодействие с цветом материала.

После обработки данных всех вершин происходит сборка геометрических примитивов и их растеризация. Следующим этапом является работа фрагментного шейдера.

» **Фрагментный шейдер [fragment shader]** – это программа, выполняемая графическим процессором и обрабатывающая отдельные фрагменты, полученные при растеризации фигур. Подобно вершинным, фрагментные шейдеры также могут быть запущены параллельно.

Список возможных операций фрагментного шейдера:

» Доступ к текстурам и их наложение на модель;
» Наложение цветов;
» Масштабирование и смещение пиксела;
» Создание эффекта тумана.

Особенности GLSL

Программистам на C/C++, наверное, будет приятно узнать, что создатели GLSL отталкивались именно от их любимцев. Действительно, большинство правил синтаксиса, равно как и построения логических конструкций, напоминают указанные выше языки. Константы, операторы, выражения и предложения – понятия одинаковые и для Си, и для GLSL. Естественно, не стоит забывать, что шейдерный язык создавался исключительно для реализации графических алгоритмов и имеет свои характерные особенности.

Во-первых, забудьте об использовании строковых и символьных значений. GLSL – язык работы только с числами! Не поддерживаются короткие и длинные целые, а также беззнаковые. При указании значения переменной необходимо следить за соответствием типов. Так, выражение float a=0; является неверным и вызовет ошибку компилятора. Правильно будет: float a=0.0;.

В языке шейдеров отсутствуют какие-либо функции для работы с файлами и побитовые операции. Зато, по сравнению с C/C++, появились новые типы данных. Для векторов могут использоваться числа с плавающей запятой, булевы и целые. Так, для первых они называются: vec2, vec3, vec4 (два, три, четыре числа). Имеется доступ к отдельным значениям вектора – либо с помощью индексации, либо по именованным полям. К примеру, значения отдельных координат можно получить, присоединив к переменной .x, .y, .z. Допускается и смешивание: .xy.

Для работы с текстурной памятью был создан специальный тип переменной (дискретизатор), характерный для конкретного типа текстурной карты: sampler1D – доступ к одномерной текстурной карте; sampler2D – доступ к двухмерной текстурной карте, и т.д.

Существуют особые спецификаторы для управления входными и выходными данными шейдеров:

» **attribute** – переменные этого типа хранят часто изменяющиеся значения и используются для передачи данных от приложения к вершинному шейдеру. Доступны только для чтения. Массивы и структуры описывать ими нельзя;
» **uniform** – может содержать относительно редко изменяющиеся данные и использоваться для обоих типов шейдеров. В отличие от атрибутов, данный описатель можно задействовать для переменных всех типов. Заметьте, что имеется некоторое ограничение на количество имеющихся uniform-переменных, накладываемое реализацией GLSL и конкретным графическим ускорителем;
» **varying** – служат исключительно для передачи интерполированных данных от вершинного шейдера к фрагментному, причем фрагментный шейдер (в отличие от вершинного) не может изменять значение полученной varying-переменной. Эти описатели должны быть объявлены в обоих шейдерах и иметь одинаковые типы. Таким образом, как и uniform, они являются глобальными и объявляются до первого использования.

Естественно, предоставляется возможность работы с матрицами – наиболее удобным типом при выполнения линейных преобразований для чисел с плавающей запятой. Поддерживаются таблицы от 2x2 до 4x4 с соответствующим обозначением mat2...mat4. Работать с ними можно как с массивом, выбирая столбцы с помощью индексации.

Помимо сказанного, GLSL имеет встроенные переменные для доступа к состояниям OpenGL. Все они начинаются с префикса gl_. Так, для вершинного шейдера имеются переменные, позволяющие, к примеру, отследить текущее состояние источника света – gl_LightSource[номер источника], gl_Fog.color – цвета тумана, и т.д.

Язык шейдеров предоставляет в помощь программисту большой набор встроенных функций, таких как: тригонометрические (синус, косинус, тангенс...); геометрические (нормализация, нахождения расстояния и длины и др.); общие математические операции (округление, модуль числа и т.д.); специализированные для фрагментного шейдера; функции доступа к текстурной памяти и многие другие. Пользоваться ими не только можно, но и нужно. Дело в том, что большинство предлагаемых функций могут обрабатываться на аппаратном уровне GPU и существенно ускорить вычисления. К сожалению, не всем функциям обеспечена такая поддержка, но их применение является хорошим тоном, хотя бы потому, что производители оборудования не стоят на месте, а значит, можно ожидать появления этой поддержки в будущем.

Первые шаги в неизведанное

Начнем с того, что каждый шейдер должен иметь свою функцию main(), объявленную как тип void – стандартная конструкция Си:

```
void main ()
{
        ...
}
```

Попробуем написать простейший шейдер, который будет выводить объект и закрашивать его в красный цвет. Распределим задачи:

1. Вершинный шейдер – трансформация вершины и запись полученного значения в глобальную переменную gl_Position;
2. Фрагментный шейдер – заливка цветом полученного пикселя через переменную gl_FragColor.

Таким образом, получается, что вершинный шейдер у нас должен выполнять стандартный функционал, а вот фрагментный подвергается необходимым изменениям. Займемся сначала первым.

```
//GLSL vertex shader
 void main ()
 {
        gl_Position = ftransform();
 }
```

Ftransform() как раз и является той функцией, что позволяет частично замещать «затертый» функционал стандартного шейдера. Она возвращает результат трансформации вершин, выполняя операции того же плана, что и базовый функционал. Учтите, что все остальные преобразования – скажем, такие, как обработка освещения – произведены не будут. Но использование ftransform() гарантирует, по крайней мере, выполнение стандартного потока операций с использованием оптимизации на уровне GPU.

Для записи цвета обработанного пикселя во фрагментном шейдере служит переменная gl_FragColor, которая, так же как и gl_Position, является заранее объявленной. Тип переменной – vec4.

```
//GLSL fragment shader
 void main ()
 {
        gl_FragColor = vec4 (1.0,0.0,0.0,1.0);
 }
```

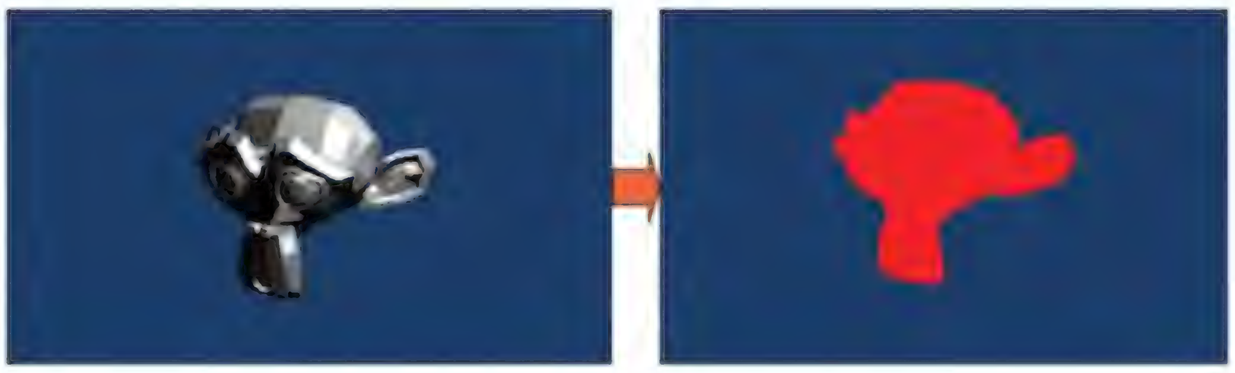
Обратите внимание на значения, передаваемые переменной. Это не что иное, как раскладка цветовой палитры RGBA (red, green, blue, alpha) [красный, зеленый, синий, прозрачность]. Как уже говорилось ранее, GLSL позволяет получить доступ к отдельным значениям вектора с помощью именованных полей. Для этой цели, в данном случае, имеются зарезервированные буквы r, g, b, a. Например, фрагментный шейдер можно было написать так:

```
//GLSL fragment shader
 void main ()
 {

gl_FragColor.r = 1.0;
gl_FragColor.gba = vec3 (1.0, 0.0, 1.0);
 }
```

Вот и все! Для проверки написанного кода просто загрузите с **LXFDVD** файл **les1_1.blend** и запустите движок клавишей P (**рис. 2**).

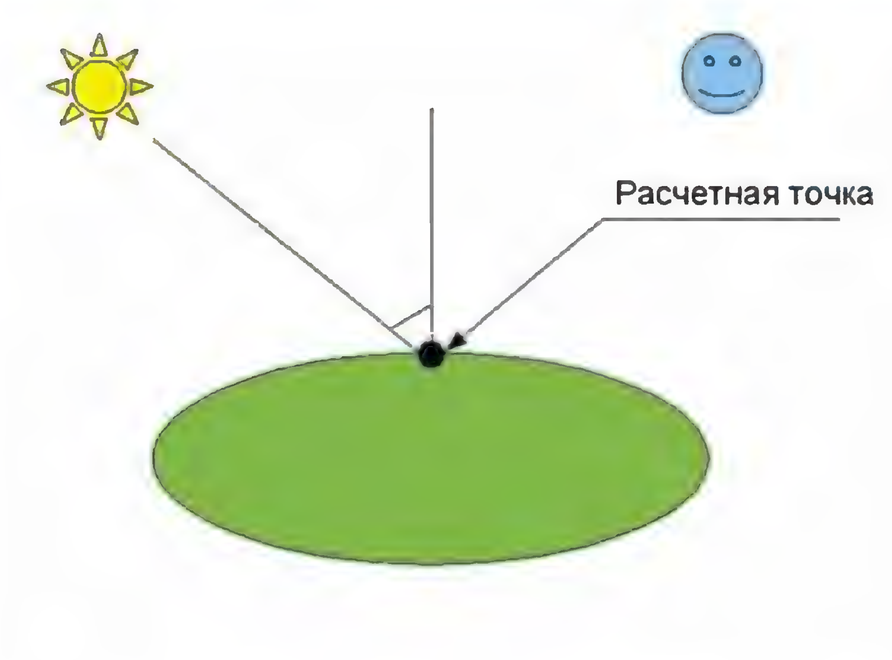
› Рис. 2. Без теней, без света, но зато с использованием шейдеров (слева – стандартный рендеринг в Blender, справа – результат работы фрагментного шейдера)!



Действительно очень просто; но не очень зрелищно. Картинка будет смотреться гораздо выигрышнее, если научить шейдер реагировать на освещение модели. Существует немало способов и алгоритмов расчета освещения, и в дальнейшем этой теме будет посвящена целая статья. Сейчас же давайте рассмотрим самый простой способ работы со светом, а именно – диффузную модель освещения, когда луч, падающий на произвольную точку поверхности, равномерно рассеивается по всем направлениям (в рассматриваемом примере расчетная функция сознательно упрощена). Все необходимые вычисления будет производить вершинный шейдер.

Вначале необходимо вычислить координаты вершины в пространстве обзора. Для этого берем локальные координаты из переменной gl_Vertex и умножаем их на текущую матрицу объекта. Заметьте, так как один из операндов – матрица, а другой – вектор, то происходит математическое, а не покомпонентное умножение:

```
vec3 position = vec3 (gl_ModelViewMatrix * gl_Vertex);
```



› Рис. 3. Векторы для расчета освещения с помощью диффузной модели.

Для вычисления самого рассеивания света нужно определить угол между нормалью к поверхности и лучом света (**рис. 3**). Добиться этого можно, если взять значения нормали и преобразовать их с помощью матрицы gl_NormalMatrix. Полученный вектор необходимо привести к единичной длине с помощью встроенной функции normalize:

```
vec3 norm = normalize (gl_NormalMatrix * gl_Normal);
```

Осталось только построить вектор из заданной точки до источника освещения:

```
vec3 lightvec = normalize (vec3 (lightPos) - position);
```

Для окончательного результата вычисления введем переменную varying outColor, которая будет хранить интерполированное значение цвета для фрагментного шейдера. Переменные этого типа должны быть определенны в обоих шейдерах одинаково. Помимо нее, нам еще понадобится константа inColor для хранения оригинального цвета объекта и uniform-переменная – для координат источника света. Следующая строка выполняет необходимые нам вычисления:

```
outColor = inColor * (max (dot (norm, lightvec), 0.0));
```

Так как может сложиться ситуация, когда источник света расположен за самим объектом, то применяется функция max для обеспечения нулевого значения рассеянного отражения при угле больше 90 градусов между направлением освещения и нормалью к поверхности. Вторая неизвестная вам функция dot выполняет скалярное произведения двух векторов.

Осталось выполнить трансформацию вершины, но на этот раз мы не будем вызывать функцию ftransform(), а выполним непосредственное перемножение координат вершины на проектную матрицу OpenGL (результат практически тот же):

```
gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
```

И вот что в итоге у нас получилось:

```
//GLSL vertex shader
 void main ()
const vec4 inColor = vec4 (1.0, 0.0, 0.0, 1.0);
uniform vec3 lightPos;
varying vec4 outColor;
{
        vec3 position = vec3 (gl_ModelViewMatrix * gl_Vertex);
        vec3 lightvec = normalize (vec3 (lightPos) – position);
        vec3 norm = normalize (gl_NormalMatrix * gl_Normal);
        outColor = inColor * (max (dot (norm, lightvec), 0.0));
        gl_Position = gl_ModelViewProjectionMatrix * gl_Vertex;
        gl_Position = ftransform();
}

//GLSL fragment shader
```

```
varying vec4 outColor;
void main ()
{
gl_FragColor= outColor;
}
```

Как видите, ничего сложного в работе с шейдерами нет, особенно когда многими учеными мужами уже были выведены необходимые нам формулы. Однако на следующем уроке мы займемся по-настоящему тяжелой и интересной работой с так называемыми процедурными текстурами. До встречи! LXF

› Рис. 4. Диффузная модель освещения в действии.

А как это в Blender?

В большинстве случаев для программирования шейдеров можно обойтись предлагаемой ниже заготовкой:

```
##-----------------------------------------------------------------
## Shader Template.py
##-----------------------------------------------------------------
import GameLogic
ObjectList = GameLogic.getCurrentScene().getObjectList()
# ---------------------------------
ShaderObjects = [ObjectList['OBCube']]
MaterialIndexList = [0]
##-----------------------------------------------------------------
# -----------------------------------------------------------------
VertexShader = """
void main() {
  //shader
}
"""

FragmentShader = """
void main() {
  //shader
}
"""

def MainLoop ():
    for obj in ShaderObjects:
        mesh_index = 0
        mesh = obj.getMesh(mesh_index)
        while mesh != None:
            # for each material in the mesh
            for mat in mesh.materials:
                if not hasattr(mat, "getMaterialIndex"):
                    return
                mat_index = mat.getMaterialIndex()
                # find an index from the list
                found = 0
                for i in range(len(MaterialIndexList)):
                    if mat_index == MaterialIndexList[i]:
                        found=1
                        break
                if not found: continue
                shader = mat.getShader()
                if shader != None:
                    if not shader.isValid():
                        shader.setSource(VertexShader,
FragmentShader,1)
                        # set uniforms

            mesh_index += 1
            mesh = obj.getMesh(mesh_index)
## call it
MainLoop()
```

Рассмотрим ее подробнее. Первые две строки подключают необходимую библиотеку BGE и загружают в переменную ObjectList список существующих объектов в сцене. Все это стандартно, но обратите внимание на третью строку в листинге. Именно в ней указывается объект, для которого будет применяться шейдер. В моем тексте это 'OBCube'. Необходимое имя объекта вы можете взять из меню 'Object and Link ', нажав F7.

Строки 'VertexShader' и 'FragmentShader' указывают на начало программных блоков, написанных на GLSL. Все передаваемые в шейдер переменные uniform располагаются после строки '# set uniforms'. Список функций для имеющихся переменных GLSL таков:

Тип переменной	Функция инициализации	Комментарий
uniform vec1 (float)	setUniform1f (name, a)	Целочисленное значение
uniform vec2 (float)	setUniform2f (name, a,b)	
uniform vec3 (float)	setUniform3f (name, a,b,c)	
uniform vec4 (float)	setUniform4f (name, a,b,c,d)	
uniform vec1 (int)	setUniform1i (name, a)	Число с плавающей запятой
uniform vec2 (int)	setUniform2i (name, a,b)	
uniform vec3 (int)	setUniform3i (name, a,b,c)	
uniform vec4 (int)	setUniform4i (name, a,b,c,d)	

Приведу пример. Пусть в вершинном шейдере используется переменная uniform vec3 lightPos. Соответственно, для ее установки служит следующая строка на Python: shader.setUniform3f('lightPos', 0.2,0.2,0.2).

А как это в OpenGL?

Для создания и управления шейдерами в OpenGL имеется специальный набор функций, предоставляемый расширениями ARB_shader_objects, ARB_vertex_shader, ARB_fragment_shader. В целом, последовательный механизм действий выглядит следующим образом:

1. Создание пустого шейдерного объекта функцией glCreateShaderObjectARB (Glenum тип шейдера). Пример:

```
testVS = glCreateShaderObjectARB (GL_VERTEX_SHADER_ARB);
testFS = glCreateShaderObjectARB (GL_FRAGMENT_SHADER_ARB);
```

2. Передача исходного кода шейдеров функцией glShaderSourceARB. Пример:

```
glShaderSourceARB(testVS, 1, &testvertex, NULL);
glShaderSourceARB(testFS, 1, &testfragment, NULL);
```

3. Компиляция каждого из шейдеров:

```
glCompileShaderARB(testVS);
glCompileShaderARB(testFS);
```

4. Создание программного объекта и присоединение шейдеров к нему:

```
shaderProg = glCreateProgramObjectARB();
glAttachObjectARB(shaderProg, testVS);
glAttachObjectARB(shaderProg, testFS);
```

5. Компоновка функцией glLinkProgramARB:

```
glLinkProgramARB(shaderProg);
```

6. Установка программного объекта как текущего:

```
glUseProgramObjectARB(shaderProg);
```

7. Установка, если нужно, начальных значений переменных uniform:

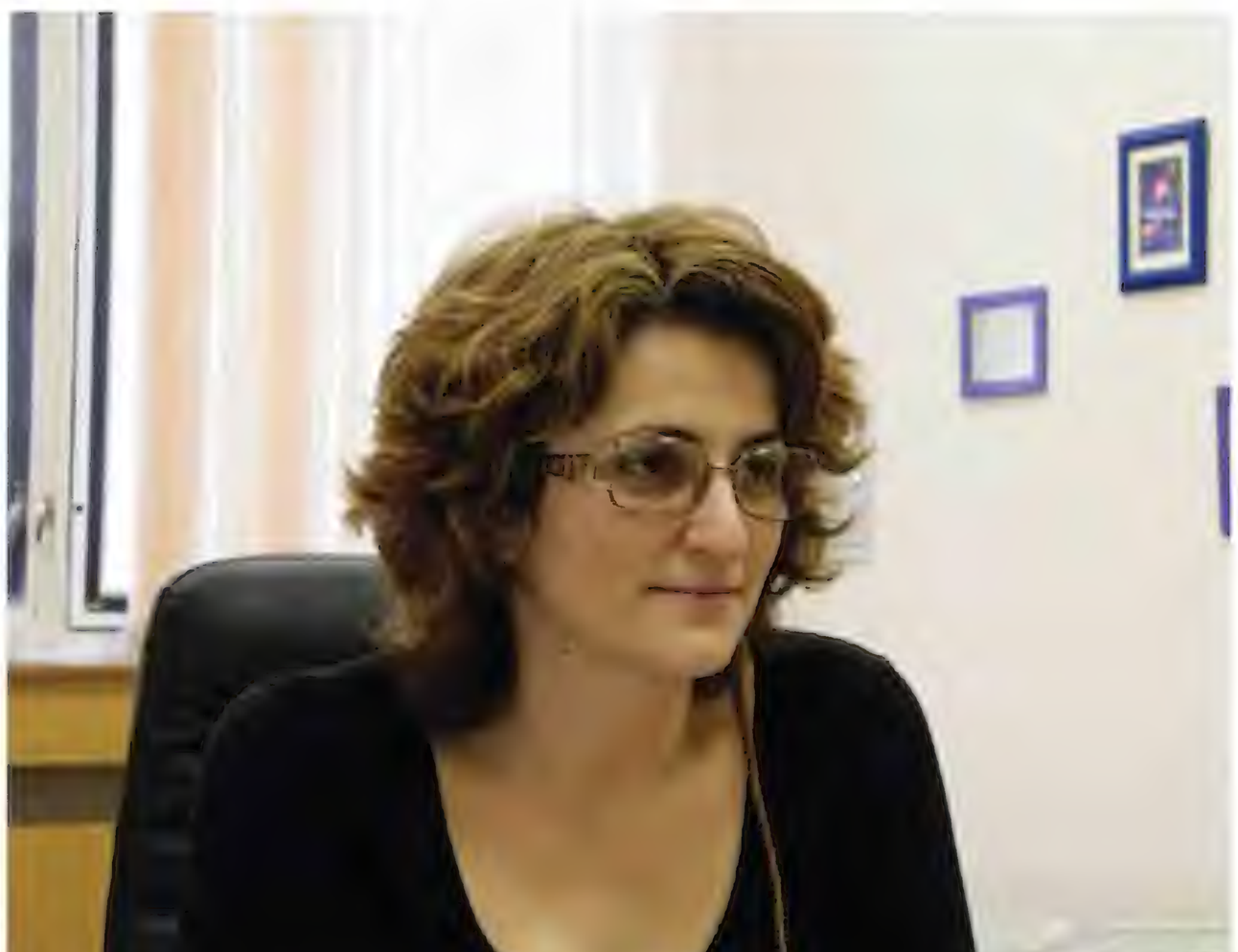
```
glUniform3fARB(getUniLoc(shaderProg, 'name'), первое значение, второе значение, третье значение);
```

» В следующий раз Процедурные текстуры через призму GLSL. Не пропустите!

Чему учиться учителю?

Начался новый учебный год – завершающий этап в процессе апробации свободного ПО в школах. Как перейти от первого знакомства к длительной и плодотворной работе? Рассуждает **Галина Пожарина**.

«...здесь, знаешь ли, приходится бежать со всех ног, чтобы только остаться на том же месте,
а чтобы попасть в другое место, нужно **бежать вдвое быстрее**. Побежали?!»
Льюис Кэрролл, "Алиса в Зазеркалье"



Наш эксперт

Галина Пожарина
Директор Центра образовательных инициатив «Открытое ПО» (г. Санкт-Петербург), PhD/DCS

Образование работает на будущее. Забегать немного вперед стандартных и привычных схем учителю не только можно, но и должно. Апробировать новый программный продукт или технологию, оценить возможности и их потенциал в ближайшей или более отдаленной перспективе – настоящее удовольствие и драйв как для ребенка, и для взрослого. Потому что это интересно.

Новость о презентации интерактивной поверхности Sphere (продукт Surface приобрел форму сферы) навела на размышления о том, что в ближайшие несколько лет изучение глобуса может стать значительно более занимательным в том случае, если статичные картинки будут заменены на Google Earth. Актуалии развития рынка субноутбуков показали их достаточно высокий потенциал для использования в учебном процессе в качестве доступной и несложно модулируемой хрестоматии – набора текстов, учебников, встроенных учебных и тестовых программ. Очевидно, что методическая и педагогическая мысль будет развиваться одновременно с этими и другими интересными технологическими решениями.

С данной точки зрения, открытое программное обеспечение – интереснейший объект. Приучить себя к его систематическому анализу современному преподавателю просто необходимо.

Интерактивные доски, электронные глобусы и микроскопы – внедрение этих технологий основано на серьезных финансовых вложениях, а потому займет достаточно большой промежуток времени, тогда как программное обеспечение, особенно открытое, свободно распространяемое, доступно для использования без проволочек и ожиданий. И здесь методические находки могут быть реализованы незамедлительно.

Накопление репозитория методических решений на основе открытых программных продуктов оптимально вести в условиях постоянно действующих педагогических семинаров в течение уже начинающегося учебного года. **Хочется напомнить, что данный (2008–2009) учебный год является завершающим в процессе апробации открытого и свободного ПО в образовании.**

Пакеты «Первая помощь» обеспечивают лицензионную поддержку программных продуктов в российских школах только до 2010 года, а значит, уже совсем скоро администрациям образовательных учреждений придется принимать самостоятельные решения об использовании программного обеспечения как в учебном процессе, так и в остальных административно-хозяйственных аспектах своей деятельности. И для принятия грамотных решений потребуется обоснованная, квалифицированная позиция. Именно поэтому различные формы повышения квалификации, связанные с Open Source, в ближайшее время должны стать очень значимыми и востребованными.

Тематика повышения квалификации может быть достаточно разнообразной:

» Перспективы развития сектора открытого программного обеспечения;
» Миграция на открытое программное обеспечение учреждений образования;
» *OpenOffice.org* – пакет прикладных программ общего назначе-ния;
» Знакомство с дистрибутивами Linux – открытой операционной системы;
» Прикладное открытое программное обеспечение для учебного процесса (по группам предметов);
» Прикладное открытое программное обеспечение для организации административной работы учреждения образования;
» Электронные библиотеки научно-методической литературы по вопросам использования открытого программного обеспечения ;
» Библиотеки (репозитории) открытого программного обеспечения;
» Электронные порталы сообщества пользователей открытого программного обеспечения;
» Открытое программное обеспечение и дистанционное обучение.

Формы повышения квалификации также могут варьироваться – от традиционных курсов до мастер-классов и семинаров. Учитывая специфику открытого программного обеспечения и его достаточно частую обновляемость, приобретает смысл повышение квалификации преподавателей и технического персонала в рамках постоянно действующих семинаров.

Просветительская работа в области Open Source ведется не только в пилотных регионах, география которых расширяется (подробную информацию см. на сайте linux.armd.ru). Центр образовательных инициатив «Открытое ПО» (www.pointfree.ru) организует семинары по Open Source для преподавателей Санкт-Петербурга и Ленинградской области. Разработки и новинки в области открытого и свободного программного обеспечения обещают быть широко представленными и на традиционных осенних выставках SOFTOOL-2008, ИНФОТЕХ-2008, Всероссийском форуме «Образовательная среда 2008».LXF

КОРПОРАТИВНЫЕ СЕРВЕРЫ
И СИСТЕМЫ ХРАНЕНИЯ ДАННЫХ
(812) 327-5960
(495) 232-9230
info@trinitygroup.ru
TRINITY
CORPORATE IT PROJECTS
Серверы
под Linux
FreeBSD
Solaris x86
для баз данных, интернет шлюзов, WEB-приложений, кластеры для научных расчетов
Анализ существующей IT инфраструктуры
Разработка технического задания
Проектирование, монтаж, внедрение
Комплексное управление IT инфраструктурой
Катастрофоустойчивые решения
Мы делаем бизнес успешным
www.trinitygroup.ru
Информационные технологии от экспертов

» Записал
Евгений Балдин

Если гора не идёт к Магомету...

Сейчас многие ругают школьное образование – и тому есть объективные причины. Но если они так напрягают, то почему бы не закатать рукава и не исправить это безобразие, как поступил **Дмитрий Леонидович Кругликов**?

С самим предметом «Информатика» я познакомился в 1981 году: в нашей школе обкатывали экспериментальную программу по математике, а заодно и по информатике. Таким образом, свою первую программу я написал как раз в то время, когда первый Apple собирали в гараже. Но честно признаюсь: программистом так и не стал. Системным администратором, полагаю, стал – сети строил. А вот программированием занимаюсь только когда никто не видит – это у меня как хобби. В 1999 году заочно окончил Харьковский государственный университет радиоэлектроники по специальности «сети связи».

С Unix-системами познакомился в 1992 году, когда на моё предприятие (Полтаваоблэнерго) «пришла» министерская сеть «Энергия». Это была BSDI [коммерческая версия BSD Unix, разработанная компанией Berkley Software Design, Inc, – *прим. ред.*]. Версию уже не помню. Хорошие учителя «подобрали» меня в то время. Я как связист помогал строить сеть передачи данных, а меня

натаскивали в администрировании. А потом как-то вспомнилась школьная подготовка, и всё быстро встало на свои места.

Сейчас проживаю на Украине в городе Полтава. Работаю в НПО «Бизнес Реинжиниринг Груп» (г. Киев). Моя специализация – консультации по вопросам миграции и построению оптимальных систем. Кроме основной работы, провожу эксперимент по преподаванию информатики (параллель 9-х классов) в школе № 17 (сейчас это специализированная гимназия).

Преподавателем я стал по собственной инициативе. Надоело смотреть на бардак в нашей системе образования вообще и в информатике – в частности. Недавно сын школу окончил (сейчас студент 3-го курса). Приходилось постоянно его подправлять, настраивать на путь истинный, так сказать… А тут дочка подросла. И ей так же «светило» изучать информатику по «традиционной» программе. На моем домашнем компьютере уже давно установлен ALT Linux, дети привыкли и спокойно работают. Как-то дочка и пошутила насчёт того, что ей придётся учить Windows, а она к этому морально не готова. В каждой шутке есть только доля шутки… Решил рискнуть и написал письмо в мэрию города – предложил свои услуги как консультанта по вопросам миграции. Получил стандартно-глупую отписку из аппарата и совершенно неожиданное приглашение из ГорОНО выступить перед преподавателями школ: поговорили и решили провести эксперимент. Естественно, что первой на него согласилась пойти наша родная гимназия. Спасибо за это администрации школы и преподавателю информатики, Валентине Владимировне Буряк, которая оказала огромную методическую помощь в составлении плана уроков. Вот так и начали.

Это решение было скорее эмоциональным, нежели рассудочным. Просто надоело быть «дойной коровой». Захотелось что-то сделать, кому-то что-то доказать… Как-то в одно время и в одной голове сложилось негативное и позитивное. Опять же, началось «дело Поносова»… Обидно стало, что кто-то имеет наглость открыть рот на человека, которого само государство ставит в такие неразумные рамки. С другой стороны, учитель, который учил моего сына, достоин того, чтобы быть защищённым от подобных выпадов. А дети вообще не должны смолоду на «иглу попадать»… Поэтому можно рассматривать преподавание как жизненную позицию. Я хочу, чтобы подрастающее поколение получило действительно лучшее, а не «самое вкусное» из «суррогатов McDonald's». *Преподавание для меня – это и хобби, и жизнь, и работа*.

Когда я рассказал владельцу фирмы о своих идеях и планах, то получил полную поддержку, время на проведение занятий и благословение – это очень помогло. Уроки в школе я провожу в рабочее время, которое оплачивается компанией: можно рассматривать мои лекции как спонсорскую поддержку.

В подобных начинаниях всегда следует иметь в виду, что преподаватели в первую очередь смотрят в программу, а потом в план лекций. Если учитель не знает, как можно изложить материал темы с использованием Linux, ему необходимо подсказать. Вся проблема в том, что преподаватели пока ещё сами не готовы сменить инструментарий.

Для выхода из этой ситуации и был организован ресурс **moodle.altlinux.ru**. Там планируется разместить и тексты лекций, и презентации, и, самое главное, методику преподавания, т.е. создать курсы для преподавателей. Заодно и программу немного изменить не помешало бы, а то она сейчас какая-то «непоследовательная»… Постоянно появляется некий «бог из машины» – технология, которая ещё не изучена, но уже применяется.

Перехода как такового не было. Просто один программный пакет поменялся на другой, а сами теоретические основы информатики остались незыблемыми. Другой вопрос в том, что пришло время вспомнить эти самые теоретические основы, которые почему-то забывают изучать, заменив их набором перемещений мыши и последовательностью «кликов» в одном, строго определённом интерфейсе.

Дело доходило до смешного. Не знаю уж, с чьей лёгкой руки в преподавании основ информатики перепутаны все термины и понятия. Например, «текстовый документ» и «текстовый файл». Если не верите, то попробуйте открыть .doc текстовым редактором… Учтите, что Word таковым не является. Так и спотыкались на каждом шагу…

В качестве базового дистрибутива я выбрал ALT Linux, так как сам давно его использую. Дабы не провоцировать религиозных войн, сразу скажу, что мне просто так удобнее. Я лучше знаю этот дистрибутив, и то, что кому-то кажется недостатком, у меня подобного отношения не вызывает. Иначе говоря: «Я просто умею его готовить». А если не умею, всегда есть кому задать вопрос. Очень хорошая команда и очень отзывчивое сообщество. Мне лично удобно и комфортно общаться с этими людьми, и я уверен в поддержке. Опять же, дистрибутив ALT Linux Junior специализирован для школ. Грех не использовать, тем более что получился он весьма качественным.

Программу и тексты лекций я пишу сам. Черновики размещаю на **moodle.altlinux.ru**. Этого ж ещё никто не делал. Точнее, разрозненные материалы в Сети есть, но преподавателю крайне сложно собрать их в единое целое. Мне проще, так как большинство технологий, приёмов и методов я на своей шкуре испытал. Многие из них зарождались и развивались на моих глазах, а об истоках некоторых понятий и терминов уже начали забывать. А я ещё помню, и многие из сообщества тоже помнят.

Основная, на мой взгляд, проблема, с которой я столкнулся – это «неправильная» учебная программа. Вместо основ информатики у нас готовят пользователей ПК, причём с первого дня приучая к явно указанному сверху набору ПО. При этом учителей *заставляют* проходить курсы по продуктам известной всем фирмы. За государственный счёт, естественно. Для того, чтобы успешно эти продукты *РЕКЛАМИРОВАТЬ*. Вот в этом и заключается главная трудность.

Из Министерства регулярно присылают разнарядки на выполнение учащимися определённых творческих работ. Для этого требуется ПО, которое стоит чуть ли не дороже, чем вся школа. Как обходить такие проблемы, я, если честно, не знаю… Посадить бы на скамью подсудимых хотя бы одного автора такой «указявки»… Как временную меру пробую составить программу так, чтобы к концу первого этапа обучения школьники владели, как минимум, навыками подготовки презентаций и рефератов в *OpenOffice.org*.

Сами дети на перемены реагируют очень по-разному. Красиво, конечно, нравится. Но и вопросы «А зачем мне это нужно, если у меня дома…, или на будущей работе…, или у папы с мамой?» тоже бывают. Правда, в последнее время все меньше и меньше. Ученики начали понимать суть своих действий, вместо «координатно-шаговой» стратегии общения с компьютером. Хотя скептики ещё остались и, думаю, останутся. Они ведь уже взрослые и сами умеют думать.

Радует то, что равнодушных практически нет. Радует скорость, с которой дети осваивают основную идею. Схему «имя_программы – набор_параметров – перечень_аргументов» осознали один раз, теперь сами видят: что, где и для чего.

Интерес к моим начинаниям довольно сильный. После открытого урока в нашей гимназии меня пригласили выступить перед учителями информатики из других школ области, в рамках курса повышения квалификации. Вот где мне досталось… Больше 4-х часов перекрёстного допроса. Вывернули меня наизнанку и выведали даже то, чего я сам не знал. Очень хорошо пообщались, порадовала заинтересованность и профессиональный интерес. Именно как к инструменту, а не просто к ОС и набору программ. В нашей же школе многие преподаватели задают вопросы о возможности применения Linux для демонстрации своих материалов.

В завершение эксперимента мы пригласили в школу преподавателей информатики города и отчитались перед ними. Мои ученики показали, чему они научились в ходе курса. Как оказалось, ничему «фантастическому»… Они просто выполнили в Linux все дей-

ствия, которые должны уметь выполнять. Все, без исключений... Восторженное придыхание вызвала разве что работа в консоли, когда одной командой делается то, на что в традиционной среде требуется минут 20.

К слову, мэр нашего города подарил школе интерактивную доску [продукт SMART Technologies Inc – www.smartboard.ru. Драйвера для Linux предоставлены изготовителем, – *прим. ред.*] Изображение на неё передаётся через проектор, а сама доска становится манипулятором. Нажал пальцем – событие обработалось: очень удобно и наглядно. Настольный планетарий на такой доске – просто фантастика.

Сыграло свою роль и открытое сообщество. Для начала, существование дистрибутивов – это результат его работы. Можно ли получить ответ на вопрос? Категорически, «да», причём как от создателей дистрибутива, так и от опытных пользователей. И сам я, как составляющая того же сообщества, пытаюсь помогать в меру сил.

Что вообще может сделать сообщество? Если коротко – все, вплоть до исправлений в исходном коде программы, которая именно вам и именно сейчас до смерти нужна, но не работает. Мне однажды в течение 30 минут такое написали... но тут нужно уметь задавать правильные вопросы.

У меня такое ощущение, что пора уже как-то прорываться на уровень нашего правительства. В России вот уже приняты хоть какие-то решения по данному вопросу, а у нас пока тишина. А время уже пришло. Ещё хотелось бы открыть в городе консультационный центр. И на вопросы частных лиц отвечать, и учителям помогать, и оказывать платные консультации по миграции для коммерческих предприятий. Обидно, когда на деньги наших граждан создаются рабочие места в других странах. Лучше собрать команду (и это будут не худшие представители) ИТ-специалистов и дать им возможность заработать. А коммерческие предприятия смогли бы сэкономить деньги. Опять же, нашли бы, куда их потратить. Но это уже выходит за рамки «школьного» вопроса.

В заключение этой эмоциональной истории я хотел бы пожелать всем апологетам свободного ПО нахальства и упорства. Во-первых, нужно понять, что никто к вам не придёт и за руку в школу не поведёт. Преподаватели пока ещё мало ориентируются в вопросе, а о том, что вы можете им чем-то помочь, даже не догадываются. Если у вас есть знания и опыт, то вы должны «преподнести» их, если хотите, естественно. И в первую очередь нужно учить учителей. LXF

Анастасия Дмитриевна Кругликова

Взрослые делают все ради детей, но часто забывают поинтересоваться их мнением. LXF решил исправить эту досадную оплошность...

LXF: Анастасия Дмитриевна, раскажите, пожалуйста, о себе.

Анастасия Дмитриевна Кругликова (АДК): Родилась 22 сентября 1993 года. Сейчас учусь в Полтавской гимназии в 9 классе.

LXF: Как же вы довели своего папу до жизни такой?

АДК: Наверное, идея о «вправлении мозгов ученикам» в Linux родилась после того, как прошлым летом я подошла и выдала: «Пааап, я не хочууу учить информатику под Виндоооой!!!!!!!!!» И так как говорила я это с самым мученическим выражением, на которое была способна, то это вполне могло толкнуть его «на такую жизнь». [*улыбается*]

LXF: Вы помогаете папе на занятиях?

АДК: Ну, многое из того, что рассказывают на уроках, я и сама не знаю... А вообще, некоторые подходят ко мне и спрашивают: «А как оно, что, действительно можно сделать так-то и так-то, а почему у меня в Windows так нельзя?» Я отвечаю.

LXF: Почему преподавание информатики с использованием свободного программного обеспечения с вашей точки зрения лучше?

АДК: Я думаю, что многие ученики, когда сами будут выбирать, что ставить на свой компьютер, возьмут то, чему их учили в школе. И им же самим будет лучше, если за ПО не придется платить. Общаясь с одноклассниками, я вижу, что интерес к Linux по этой причине очень большой! Уже многие просили: «Настя (или Дмитрий Леонидович), а можете записать мне Linux? Я хочу дома попробовать.»

LXF: Как к этому относятся ваши одноклассники?

АДК: Многие действительно ставят Linux дома, хотя большинство ещё не решило, а некоторым не разрешают родители. Мнений типа «Windows намного лучше, а это мне не нравится» я пока не слышала.

LXF: Чего, с вашей точки зрения, не хватает в Linux?

АДК: Не знаю... Мне все нравится. Вообще, сложно представить, как бы я училась под Windows. Например, когда проходили поиск файлов, я дома тренировалась из командной строки и находила за пару минут всё, что мышью искала бы минут 10. В Linux, из командной строки, мне нравится больше!

LXF: Как вы относитесь к Linux-сообществу?

АДК: Ну, в принципе, Linux для меня пока просто удобный инструмент. Но очень удобный.

LXF: Что бы вы посоветовали/пожелали сообществу, чтобы внедрить GNU/Linux в школы?

АДК: Наверное, надо донести до всех, что ПО можно бесплатно качать из Интернета, и что для работы с ним не надо быть профессиональным системным администратором или программистом.

В «Системном администраторе» вы не прочтете о:

– котировках валют
– сплетнях
– погоде
– политике
– развлечениях

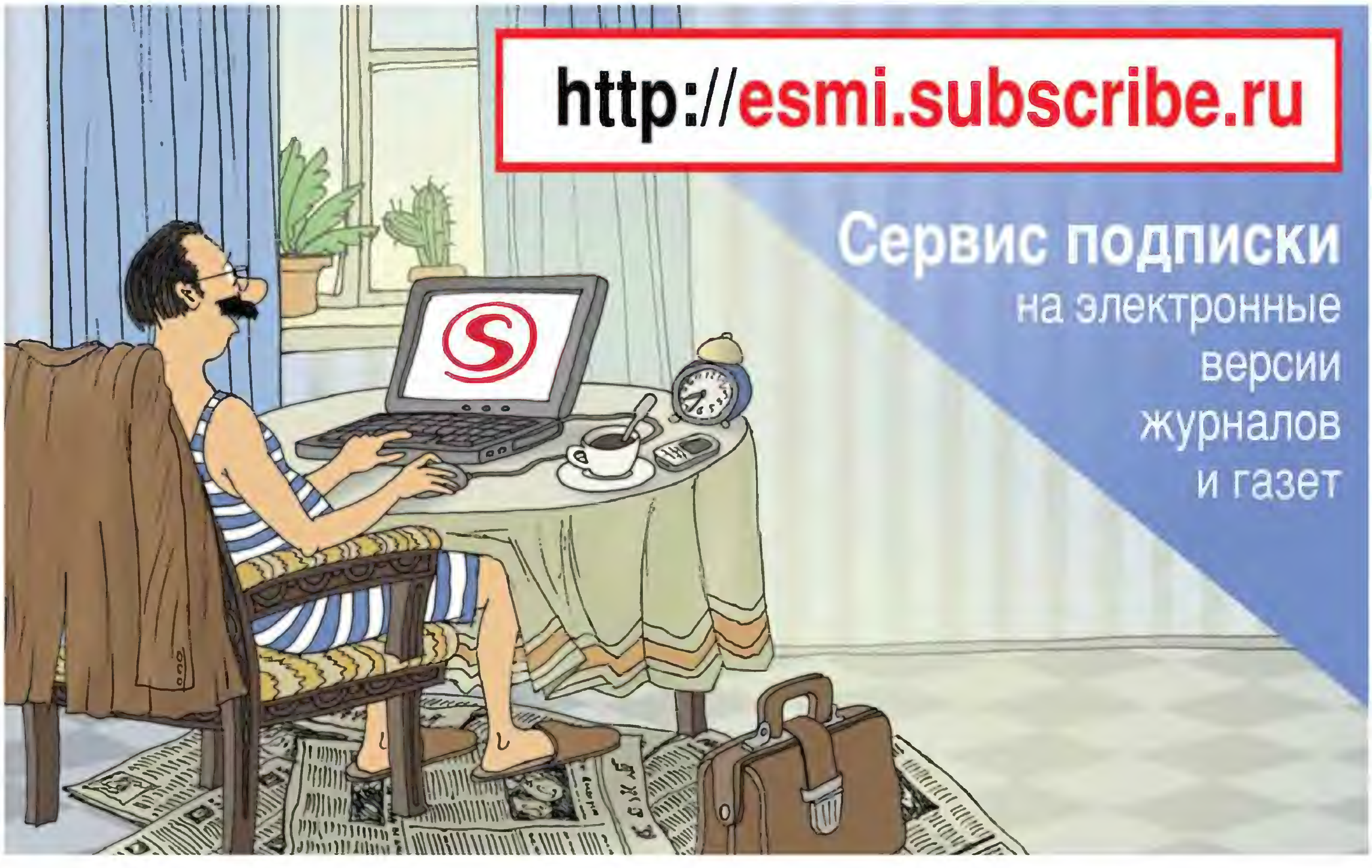
В вашем распоряжении:

– опыт лучших IT-специалистов
– новые идеи и полезные советы
– самые эффективные решения в области системного и сетевого администрирования

Подпишитесь сейчас!

Время подписки ограничено!

Роспечать – 20780, 81655
Пресса России – 87836
Online-подписка – www.linuxcenter.ru



Ответы

Есть вопрос по Open Source? Пишите нам по адресу: answers@linuxformat.ru!

В этом месяце мы отвечаем на вопросы про:

1. Установку ПО
2. Проблемы с загрузкой
3. Беспроводные сети
4. Lilo
5. Grub
6. Снова Grub!
7. Поддерживаемые «материнки»
8. Восстановление видео
9. Grub, в третий раз!
10. Контроль над компьютером
11. SD-карты

1 Установка HotPicks

Я наконец-то собрался с духом и поставил Mandriva 2008 Powerpack на свой ноутбук Toshiba (Satellite M50 PSM53A). Все, до чего я добрался, работает прекрасно, но, похоже, у меня soft-модем. Ладно, как-нибудь выкручусь.

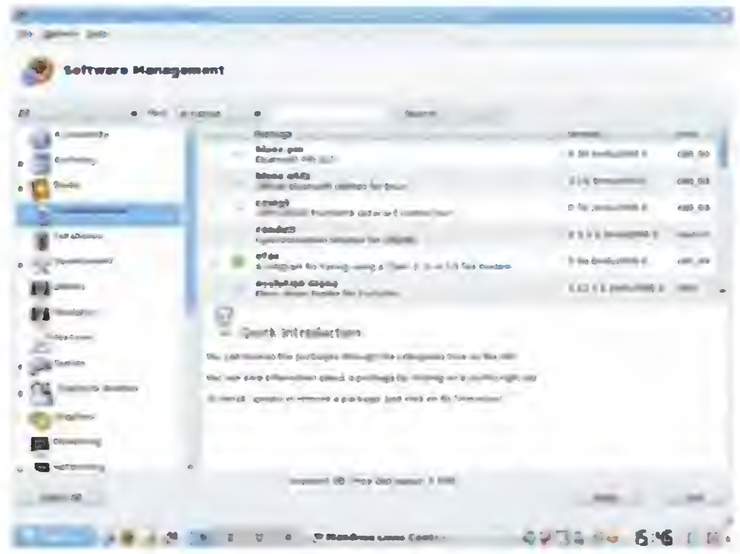
Решил установить *GCstar* и *Genius* с LXF102. Несколько бесплодных часов проковырявшись в утилите Управления ПО, руководствах и на сайте Mandriva, я пришел к выводу, что пользователям дозволено брать только предоставленные им программы [в репозиториях, – прим. пер.]. С виду, то же самое и в других дистрибутивах. Скажите мне, что я ошибаюсь! А если нет, то почему так? Как догадаться, что ПО из вашего журнала подходит для моего конкретного дистрибутива?

Кроме того, важно ли, какой рабочий стол – Gnome или KDE – у меня стоит, и как определить, для какого рабочего стола предназначено ПО? Если я установлю Gnome на Mandriva, смогу ли я переключаться между Gnome и KDE, как это делается в Fedora?

Якоб Вестерхоф [Jacob Westerhoff]

Различные дистрибутивы делают огромную работу по упрощению установки и использования программ, насколько это возможно. Для этого они поддерживают огромные репозитории, где содержится все, что Вашей душе угодно. Эти пакеты протестированы, и Вы можете быть уверены, что они совместимы с данным дистрибутивом и друг с другом. В итоге установка через менеджер пакетов обычно чрезвычайно проста, а все детали разрешения зависимостей, скачивания файлов и настройки ПО скрыты от пользователя. Создатели дистрибутивов и вправду предпочитают, чтобы программы устанавливались только из их репозиториев, но это не означает невозможность добавления ПО из других источников.

Некоторые менеджеры допускают установку из



Установка программ из собственных репозиториев дистрибутивов гораздо проще, но можно установить и стороннее ПО, например, с LXFDVD.

отдельных файлов пакетов; часто при этом они требуют манипуляций с командной строкой. В Mandriva Вы должны использовать команду *urpmi*, выполняемую в терминале от имени суперпользователя-root.

```
urpmi some-package.rpm
```

Этот способ будет работать только для пакетов в корректном формате, в зависимости от дистрибутива: RPM для Mandriva, Fedora и SUSE; Deb для Debian и Ubuntu. У него также есть то преимущество, что установленные пакеты включаются в специальную базу дистрибутива, и можно отследить, что вы уже инсталлировали. Но две упоминаемые Вами программы доступны только с DVD в виде исходных кодов, и перед установкой их нужно скомпилировать. В случае *GCstar* Вы можете прочитать, как это делается, в файле **INSTALL.txt** на DVD. Для *Genius* и большинства других программ файл, описывающий процесс установки, называется **INSTALL** и находится внутри tar-архива. Порядок действий здесь такой:

```
tar xf /media/dvd/Hot_Picks/Genius/genius-
1.0.2.tar.gz
cd genius-1.0.2
less INSTALL
./configure
make
su -c “make install”
```

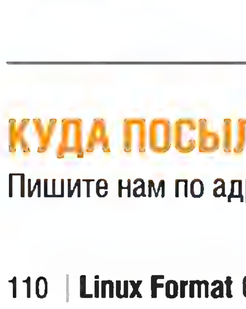
Эти команды распакуют архив, переключатся на директорию с его содержимым, отобразят файл **INSTALL**, сконфигурируют процесс сборки для Вашей системы, скомпилируют программу и установят ее, соответственно. Установка скомпилированного ПО требует прав администратора, следовательно, используется *su*. Пользователи Ubuntu должны заменить последнюю строку на

```
sudo make install
```

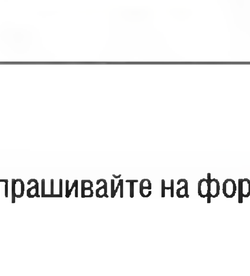
Вам также может понадобиться сначала установить *GCC* – компилятор программ. Команда **./configure** проверит наличие его и другого необходимого ПО в системе, и сообщит Вам, если чего-то не хватает.

Наши эксперты

Мы найдем эксперта по любому вопросу – от проблем с установкой системы или модемом до сетевого администрирования; главное – спросить!



Нейл Ботвик
Владелец ISP и экс-редактор дисков для нашего журнала, Нейл считает, что в Linux он от скуки на все руки.



Майк Сондерс
Майк был одним из создателей прототипа LXF – Linux Answers. Его специальности – программирование, оконные менеджеры, скрипты инициализации и SNES.

Пол Хадсон
Пол – местный супер-программист, и он может и хочет управиться со всеми вашими проблемами по части web и баз данных.

Грэм Моррисон
Когда он не обозревает кучи программного обеспечения и не халтурит с MythTV, Грэм готов дать ответ касательно любого оборудования и проблем виртуализации.

Валентин Синицын
В редкие свободные минуты главный редактор нашего журнала обычно запускает *mcedit*, чтобы отшлифовать какое-нибудь открытое приложение. Его любимая тема – настольный Linux.

Юлия Дронова
Когда ее компьютер не занят выполнением команды emerge, Юлия использует его, чтобы модерировать ЛинуксФорум.

КУДА ПОСЫЛАТЬ ВОПРОСЫ:

Пишите нам по адресу: answers@linuxformat.ru или спрашивайте на форуме: www.linuxforum.ru

Распространение в виде исходных кодов – самый дистро-независимый метод, он должен работать на любом дистрибутиве и с любой аппаратурой (но зато потребует немного дополнительных усилий). Запуск программ и KDE, и Gnome зависит от наличия подходящих библиотек, а большинство дистрибутивов содержат их, так что можно запускать программы KDE в Gnome и наоборот. Вы можете установить Gnome и KDE в Mandriva и выбирать между ними при входе в систему, но Вам не требуется переключать рабочий стол просто для запуска программ. Единственная проблема при запуске приложений KDE в GNOME та, что они выглядят «не к месту». То же самое относится и к программам Gnome в KDE, но, к счастью, существует модуль KDE для применения текущей темы рабочего стола к программам Gnome/*GTK*. **GTK-Qt Theme Engine** доступен на сайте **http://gtk-qt.ecs.soton.ac.uk**. [Советую также обратить внимание на QtCurve – универсальную визуальную тему для Gnome и KDE, с ее помощью ваши приложения будут выглядеть практически идентично на любом рабочем столе, – *прим. ред.*] **НБ**

2 Плохой IPL! Плохой!

У меня недорогой ноутбук Time AMD 64 с Windows XP и обычно Mandriva в двойной загрузке. Временами я тестирую на нем другие дистрибутивы, иногда параллельно с Mandriva, иногда временно заменяя ее. На этой неделе я попробовал установить Foresight, используя существующие разделы Linux и выбрав загрузчик по умолчанию. После рестарта я просто получил сообщение об ошибке 'Bad IPL' 'Press any key to reboot'. И больше ничего не происходит.

Google выдал мне, что IPL означает 'Initial Program Load' [Первоначальная программа загрузки] – для меня это пустой звук. Попробовал вернуть Mandriva, но это не помогло – ошибка та же. Я хочу попытаться спасти директории с моими документами, картинками и музыкой, но еще важнее для меня электронная почта и новости, находящиеся на разделе RISC OS в Win XP. У меня есть соблазн попробовать Insert (с **LXF105**), но я не уверен, что это то, что мне нужно.

Джон Малруни [John Mulrooney]

IPL – первый этап загрузчика: этот код загружает программу, которая выводит для Вас загрузочное меню. IPL размещается в главной загрузочной записи жесткого диска (MBR), где ему доступно всего лишь 446 байт – это менее половины объема, потребовавшегося Вам для описания проблемы, поэтому уж простите его за столь неинформативное сообщение об ошибке. Оно появляется, если код Вашего загрузчика поврежден, и ошибка, которую Вы видели, фактически идет от BIOS.

Вы можете переустановить загрузчик с диска в режиме восстановления. Наберите **linux rescue** в строке загрузки. Ваш Foresight должен смонтироваться в **/mnt/sysimage**, откуда Вы сможете переустановить загрузчик в MBR с помощью следующих команд:

```
cat /mnt/sysimage/boot/extlinux/mbr.bin >/dev/
sda
```

Это подразумевает, что он у Вас уже установлен, на первый (или единственный) жесткий диск. Теперь выйдите из консоли, набрав **exit**, или, если Вы лодырь типа меня, нажав Ctrl+D, и он должен корректно перезагрузиться.

Второй вариант – переключиться с *Extlinux* на *Grub*. Foresight устанавливает оба, но настраивает только загрузчик по Вашему выбору, поэтому у *Grub* имеется лишь шаблон файла меню. Если Вы уже знакомы с синтаксисом меню *Grub* или читали инструкцию об этом на **www.gnu.org/software/grub/**, можете отредактировать файл и установить его, загрузившись в режиме восстановления и войдя в установку с помощью *chroot*, например, так:

```
chroot /mnt/sysimage
nano /boot/grub/grub.conf
# отредактируйте файл и нажмите Ctrl-X для записи
и выхода
grub-install /dev/sda
```

Нажмите Ctrl+D дважды, сперва для выхода из chroot, потом для выхода из оболочки, и компьютер должен перезагрузиться с новым меню *Grub*.

Если у Вас нет желания редактировать меню *Grub*, можете просто переустановить Foresight, но на сей раз выбрать *Grub* вместо указанного по умолчанию *Extlinux*. Я бы не должен был выступать за переустановку системы как средство решения отдельных проблем, но ведь Вы не пользовались ею (потому что не могли запустить), и терять Вам нечего.

Большинство Live CD/DVD дистрибутивов позволят Вам смонтировать раздел Windows для резервного копирования Ваших данных. **МС**

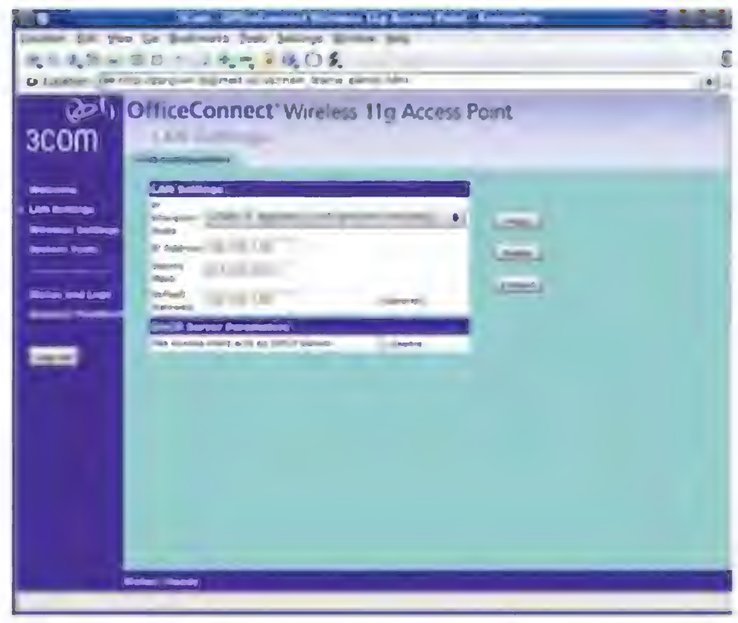
3 В сети без проводов

По совету LXF, я надсадился и купил ноутбук Dell. Не тот, что описан в статье – его уже распродали, но следующую модель, Inspiron 1525. На нем благополучно запустился Ubuntu, и это мое первое беспроводное устройство. Поэтому мне понадобилась точка доступа. У меня уже есть сеть, с установленным LAMP, также включающая брандмауэр и маршрутизатор с выходом в Интернет. Но я не знаю, как работать с беспроводной сетью. Влез на Google, прочитал отзывы и увидел много устройств, содержащих точку доступа, коммутатор и маршрутизатор или просто точку доступа, но я не могу понять, где они сидят в моей сети.

Возможно ли использование точки доступа/коммутатора типа (например) Linksys WRT54GL для расширения моей проводной сети с помощью беспроводной через коммутатор? Я знаю что это маршрутизатор, но я хочу, чтобы он был коммутатором, чтобы беспроводная сеть была в том же сегменте 192.168.2.0/24. Если я установлю его как маршрутизатор, появится дублирование NAT для работы с Интернетом, а это не позволит мне соединяться с удаленным компьютером на работе из дома. Можно ли работать через бетонный пол, чтобы беспроводная сеть расширилась на нижние этажи?

Joey C, с форумов

Вам нужна простая точка доступа. Она подключается к имеющейся проводной сети на Вашем маршрутизаторе или коммутаторе и добавляет к ней Wi-Fi-доступ. Можно использовать также устройство «все-в-одном», включающее маршрутизатор и точку доступа (и даже модем), но придется убрать лишнее, усложнив настройку по сравнению с использованием простой точки доступа [и переплатив, – *прим. ред.*]. Точка доступа будет отвечать за беспроводное соединение и шифрование, а к остальному применятся настройки Вашей проводной сети. Но вот на одну вещь следует обратить пристальное



При добавлении беспроводной точки доступа в существующую сеть вам, вероятно, придется выключить службу DHCP.

внимание: точка доступа обычно имеет встроенный DHCP-сервер. Если в Вашем маршрутизаторе он уже есть, отключите DHCP в точке доступа, так как два независимых DHCP-сервера в одной сети вызовут конфликт. Все точки доступа, которые я использовал, имели web-интерфейс (доступ к нему Вы получите из проводной сети), где Вы можете отключить DHCP.

Радиус покрытия – сложная тема: все, что плотнее воздуха, находящееся между точкой доступа и ноутбуком, до известной степени его сокращает. Обычно большинство ненаправленных антенн имеют охват только в горизонтальной плоскости, с ограниченным покрытием в вертикальной; антенна с высоким усилением увеличивает этот эффект. Подключаемая антенна – устройство направленного действия – позволит Вам отрегулировать охват по горизонтали и вертикали, но чтобы найти лучшее положение, придется экспериментировать. Поскольку для улучшения покрытия, вероятно, придется заменить Вашу антенну, убедитесь вначале, что выбранная точка доступа позволяет это сделать. Как правило, это так, но бывают и встроенные антенны. **НБ**

4 Игра в девятку

Уверен, вы помните: одна из обычных ошибок в стадии загрузки – это сбой старта ядра, что обычно происходит при его обновлении, но упущении вызова *Lilo*. Ну, *Grub* – это, конечно, другая история.

Так вот, когда ядро не найдено, по всему экрану ползут девятки. Но почему именно девятки, а не шестерки или другие цифры? Раз уж загрузчик указывает на неправильный сектор диска, не должны ли мы видеть случайные даные, хотя бы в форме чисел?

Джордж [George]

Девятка не случайна: это код ошибки *Lilo*. Когда *Lilo* загружает сам себя, он пишет на экране слово *Lilo*, по одной букве при успешном завершении каждого этапа. Если на каком-то этапе происходит сбой, *Lilo* выводит двузначное шестнадцатеричное число, идентифицирующее ошибку. Код ошибки 99 означает «неверный индексный сектор второго этапа» – другими словами, *Lilo* не может найти продолжение своего кода в Вашей файловой системе. Код ошибки повторяется, почему экран и заполняется девятками.

Для нахождения файлов *Lilo* использует не инфор-

мацию из файловой системы, а адрес физических блоков в коде загрузчика. Потому-то и надо перезапускать *Lilo* после внесения любых изменений, будь это редактирование меню или установка нового ядра: иначе загрузчик ищет этот код в неправильном месте, видит, что нужных данных там нет, и выводит сообщение об ошибке 99. Экран не показывает Вам случайную информацию, потому что *Lilo* прост, но не глуп, и он осознает, что имеет дело с некорректным расположением. Выходов, как всегда, два: честно перезапускать *Lilo* каждый раз после обновления ядра или изменения меню, или переключиться на *Grub*, который, несмотря на лаконичность кодов ошибки, по крайней мере делает их чуть более понятными. **ПХ**

5 Капризный модем

В Я рассматриваю Linux в качестве замены для Windows XP, и мне очень импонирует наличие ряда программных продуктов, однако я не могу получить доступ к Интернету. У меня внешний модем на 56k, и все дистрибутивы Linux, которые я попробовал, не смогли его распознать. Я новичок в Linux и по-настоящему не разбираюсь в этой проблеме, так что мои поиски информации ни к чему не привели. Можете ли вы указать мне правильное направление?

Джон Хил [John Hill]

О У Вас последовательный или USB-модем? Если это старый добрый последовательный [COM, – *прим. пер.*] модем, то проблема решается просто. Первый последовательный порт – это **/dev/ttyS0** (эквивалент COM1 в Windows), и надо просто настроить программу дозвона на его использование. Какую программу дозвона взять, зависит от Вашего дистрибутива и, что важнее, рабочего стола. В KDE применяется *KPPP*, в случае Gnome зайдите в Система > Администрирование > Сеть и выберите опцию Точка–точка [Point-to-Point] или Модем [Modem].

В случае USB-модема Вам может повезти, а может, и нет. USB-модемы чем-то похожи на внутренние: одни просто работают и поддерживаются ядром; другие требуют специфических драйверов, которые могут иметься только для Windows; третьи лежат между ними, и их можно заставить работать некоторыми ухищрениями. Используя команду **lsusb -v**, узнайте детали Вашего модема, затем обратитесь в Google или другой поисковик для определения совместимости устройства с Linux, чтобы выяснить, будет ли он «просто работать», потребует ли драйвер или это пустая затея.

Быстрый тест – запустить

```
tail -f var/log/messages
```

в консоли от имени root (в случае Ubuntu просто допишите в начале **sudo**) до подсоединения модема, затем воткнуть его и засечь сообщение о подключении. Если марка и модель распознались, все в порядке. Если появилось устройство **/dev/ttyUSB0**, то Вы – везунчик и можете использовать это имя устройства в программе дозвона, как с последовательным модемом. Иначе опять придется бороздить просторы Интернета. **НБ**

Для использования внешнего последовательного модема достаточно просто ввести параметры – но их необходимо где-то узнать.

6 Красиво, но бесполезно

В У меня ноутбук Sony Vaio VGN-N385N, и запуск любых дистрибутивов с KDE до 3.5.8 или любого Gnome-дистрибутива не вызывает проблем. Но все дистрибутивы с KDE 3.5.9/4.03 или новым Gnome (включая Foresight с LXF106 DVD) просто не хотят работать. Они без проблем устанавливаются на жесткий диск, но через минуту после старта процесса загрузки виснет.

Сообщения *Grub* появляются и проходят, далее идет окно с заставкой, а затем через весь экран начинает мигать цветной прямоугольник, и больше ничего! Это происходит в Ubuntu, Kubuntu, Mandriva Spring и теперь в Foresight.

Может быть, надо что-то изменить в загрузчике?

Мартын [Martyn]

О Сбой происходит слишком рано, чтобы обвинить в нем KDE или Gnome. Загрузчик тоже ни при чем: он передает контроль ядру еще до появления заставки и далее не участвует в процессе. Подозреваю, что Ваша проблема вызвана последними версиями какого-то аппаратно-зависимого системного ПО – либо ядра, либо чего-то вроде HAL и udev, не ужившегося с Вашим оборудованием. Однако это только подозрения, и Вам необходимо выяснить, что ломается, во время загрузки.

Красивые экранные заставки, применяемые в большинстве дистрибутивов, милосердно скрывают жуткий текст, выводимый системой на самом деле. Но именно этот текст почти наверняка указывает на источник Вашей проблемы. Одни дистрибутивы имеют опцию загрузки в «безопасном режиме», отключающем заставку; другие позволяют убрать ее нажатием клавиши, обычно Esc или F2. Сделав это, Вы увидите точку сбоя, что в 90% случаев позволяет ее исправить.

Если же опций ликвидации заставки нету, можно убрать ее из меню *Grub*. Нажмите E (для редактирования), когда подсвечен пункт меню по умолчанию, переместите выделение на строку, начинающуюся с ‘kernel’, и снова нажмите E. Строка kernel содержит несколько опций; нам нужны те, что отвечают за вывод заставки. Например, в Ubuntu она выглядит так:

```
kernel /boot/vmlinuz-2.6.22-14-generic
root=UUID=xxxx ro quiet splash
```

Удалите опции queit и splash, нажмите Enter для окончания редактирования и B для загрузки с измененными опциями. Некоторые дистрибутивы также содержат опцию quiet в отдельной строке. Можете отключить ее, выделив и нажав D.

Вопрос-победитель (английская версия)

Карточные фокусы

В Очень хочется купить ноутбук типа Asus Eee PC, но я озадачен использованием SD-карты для расширения памяти. На главной странице Panasonic есть программа форматирования SD-карт. Судя по всему, Windows не делает этого должным образом. А как насчет Linux? И еще: это общая проблема, или она связана только с использованием SD-карт в фотокамерах?

Педер Кристенсен [Peder Christensen]

О Для ясности отметим, что Eee может использовать карты SD как хранилище информации, но не как память (конечно, можно размещать на них раздел подкачки, но это будет очень медленно). Память увеличивается заменой модуля SO-DIMM внутри компьютера, как описано в **LXF106**. Что касается хранения данных, то SD-карты обычно продаются уже отформатированными в файловую систему FAT, но я предпочитаю форматировать их в том устройстве, где будет использоваться карта, например, в фотокамере или КПК. Проблем с чтением таких карт в Linux у меня никогда не было. Если Вы хотите использовать SD-карту как дополнительный диск в Eee PC, лучше всего будет переформатировать ее с использованием файловой системы Linux. Я бы порекомендовал ext2, так как отсутствие журналирования сокращает количество записей на карту и увеличивает доступное место.

Eee может использовать карты SDHC (Secure Digital High Capacity); на моем стоит 8-ГБ карта. Однако учтите: они отличаются от обычных SD-карт, хотя на вид такие же. Поэтому при использовании карт SDHC в Вашем Eee понадобится также совместимый карт-ридер (а большинство из них, увы, несовместимы), чтобы передавать данные с Вашего настольного компьютера. **НБ**

» Теперь Вы должны увидеть пробегающий по экрану текст, в основном неинтересный. Важны только несколько последних строк перед остановкой. Поищите в Google по последнему сообщению об ошибке или зайдите на Линуксфорум по адресу www.linuxforum.ru. Но сначала попробуйте некоторые опции загрузки: они могут решить больше аппаратно-зависимых проблем, чем все другие вместе взятые. Удалив quiet и splash из строки kernel, добавьте **noapic acpi=off**. Обратите внимание на различия в написании: хотя слова и похожи, это две очень разные опции. Если Вы загрузитесь с обеими, попробуйте их по отдельности. Найдя наилучший набор опций, можете отредактировать файл меню в **/boot/ grub/menu.lst**, сделав изменения постоянными (для этого Вам потребуются права root). НБ

7 Материнская поддержка

ВЯ запускаю OpenSUSE 10.2, но ОС не хочет нормально работать на моей новой машине с материнской платой Asus P5KPL. Какие дистрибутивы Linux будут на ней хорошо работать и можно ли на ней также запустить *Planner*?

Флориан Хубер [Florian Huber]

ОКрупные изготовители типа Asus обычно работают хорошо, благодаря многочисленности пользователей их продуктов. Однако для поддержки нового оборудования в ядре требуется время, да еще эта версия ядра должна войти в дистрибутив. Большинство дистрибутивов выходят два раза в год – вместе с циклом тестирования, предшествующим релизу, может пройти девять месяцев, прежде чем новое оборудование попадет в последний релиз вашей системы. Asus P5KPL не содержит ничего такого эзотерического, поэтому он должен поддерживаться всеми текущими дистрибутивами. Ключ к разгадке такой: OpenSUSE 10.2 был выпущен в декабре 2006 года, поэтому поддержка чего-либо моложе 18 месяцев в нем будет проблематичной. Установка более нового дистрибутива решит все Ваши проблемы. Некоторые поставщики Linux имеют на своем сайте раздел об аппаратной совместимости, где Вы можете навести справки перед скачиванием и установкой. Например, OpenSUSE имеет такой на http://en.opensuse.org/HCL, а Mandriva – на www.mandriva.com/hardware. Многие дистрибутивы предоставляют LiveCD-версию, позволяющую убедиться, что все работает, перед фактической инсталляцией.

Planner должен работать в любом дистрибутиве, имеющем библиотеки Gnome. Он есть в репозиториях OpenSUSE, и Вы можете установить его из Yast обычным путем. НБ

8 Спасите видео

ВЯ установил/обновил Ubuntu 7.10 до 8.04 и напортачил с экраном: он теперь широкоформатный и нечитаемый. Загрузка в режиме восстановления не помогла, поэтому мне нужно как-то вернуть, скажем, режим VESA 800x600.

Роэл Виссер [Roel Visser]

ОТакое возможно: зайдите в терминал и введите

```
sudo dpkg-reconfigure xserver-xorg
```

Запустится утилита настройки графического режима, где Вы сможете задать правильные параметры. Она не только поможет Вам при поломке: благодаря ей Вы можете безопасно экспериментировать с Вашими настройками, осознавая, что если Вы наведете хаос, то спасение рядом – идеальные условия для тех, кто практикует провокационные работы с техникой. ГМ

› **Управляйте вашим принтером через web-браузер с помощью CUPS** (см. ЧАВО, ниже).

9 Возвращаем *Grub*

ВПока что я постоянно перезагружаюсь между XP Pro на диске C и PCLinuxOS на диске D. Пора уже переустановить XP, но установка PCLinuxOS прекрасно настроена, и я не хочу делать все заново, и не хочу потерять двойную загрузку. Возможно ли полностью переустановить XP без потери двойной загрузки? Или есть способ восстановления двойной загрузки после установки XP?

Crow_se, с форумов

УУстановка Windows сметет любой загрузчик Linux (в случае PCLinuxOS – *Grub*). Поэтому при настройке машины с двойной загрузкой лучше установить Windows первой. Но Ваши настройки легко вернуть, потому что Windows всего лишь перезаписывает код загрузчика в главной загрузочной записи (MBR), не затрагивая конфигурацию Вашего

Часто задаваемые вопросы

CUPS

Система Common Unix Printing: пояснения

» Что такое CUPS?
Common Unix Printing System [Универсальная система печати для Unix]. Это набор драйверов и утилит для полной поддержки, управления и использования принтеров в Linux и других Unix-подобных ОС.

» Так это драйвер принтера?
Отчасти да, но это намного шире, чем просто драйвер. CUPS помещает «переносимую прослойку» между приложениями и печатающими устройствами. В нее входят и драйверы принтеров, и все необходимое для общения программ с ними.

» Значит, придется мучиться с командной строкой и файлами конфигурации, чтобы мой принтер заработал?
Вовсе нет. У CUPS есть собственные графические инструменты настройки, работающие через браузер. Зайдите на http://localhost:631 (может понадобиться Ваш пользовательский пароль или пароль root) – вы попадете на домашнюю страничку CUPS. Здесь можно просматривать, добавлять и удалять принтеры точно так же, как вы управляете задачами в очереди печати; можно даже просмотреть документацию.

» Зачем использовать браузер вместо стандартных программ с графическим интерфейсом?
Web-интерфейс означает, что вам не нужны никакие графические «инструменты», не требуется даже запускать X-сервер на Вашей машине. Вы можете получить доступ к интерфейсу через текстовые браузеры типа *Elinks* или через обозреватель на другой машине.

» А это не опасно?
Сомнения понятны, но «из коробки» CUPS допускает только подключение с локальной машины. Можно изменить настройки для доступа из локальной сети (интернет-подключение не рекомендуется), а также для контроля над правами отдельных пользователей – допустим, позволить им печатать, но не разрешить удалять принтер.

» Что ж тогда остается на долю *Gimp-Print*, и при чем тут *GIMP*?
Gimp-Print – набор принтерных драйверов, специально созданных для работы с графическим ПО *GIMP*. Хотя *GIMP* неплохо взаимодействует и с CUPS, некоторые принтеры дают лучший результат на драйверах *Gimp-Print*. Эти драйверы теперь тоже работают с CUPS, поэтому *Gimp-Print* можно рассматривать как расширенный набор драйверов, работающих через CUPS со всеми программами, не только с *GIMP*.

» А что такое *Gutenprint*?
А это новое название *Gimp-Print*. В зависимости от свежести Вашего дистрибутива, у вас будет либо *Gimp-Print 4x*, либо *Gutenprint 5.0*. *Gimp-Print* теперь предназначен не только для *GIMP* – отсюда и новое название, оно ближе к назначению пакета. Но смена имени всегда вызывает… да уже и вызвало некоторую путаницу. Например, в репозиториях некоторых дистрибутивов до сих пор сидят пакеты *Gimp-Print*.

» Как узнать, поддерживается ли мой принтер?
Обратитесь на www.linuxprinting.org. Там содержится внушительная база данных о том, насколько хорошо (или не очень) поддерживается каждый принтер, и приведены советы по использованию драйверов. Лучше посетить этот сайт до покупки принтера, тогда новая вещь уж точно не окажется бесполезной.

меню. Для восстановления двойной загрузки Вам нужно запустить *Grub*, что Вы сможете сделать с любого LiveCD-дистрибутива. Загрузитесь с CD (или DVD), откройте терминал и войдите как root, запустив **su** (или **sudo bash**, если у Вас CD Ubuntu). Вы можете выполнить автоматический скрипт установки так:

```
grub-install /dev/sda
```

Правда, это может не сработать при наличии нескольких жестких дисков.

Альтернатива – установить загрузчик вручную, тут нужны всего лишь две команды. Только определитесь с Вашим диском и разделами, потому что *Grub* использует свою схему именования. Первый жесткий диск – это **(hd0)**, первый раздел на нем – **(hd0,0)**. То есть *Grub* считает от нуля. Уточнить надо два местоположения: директории **/boot/grub**, которое может быть **(hd1,0)**, если это первый раздел второго диска, и куда установить код загрузчика – обычно **(hd0)**, если Вы хотите разместить его в MBR первого диска. Узнав, где все должно быть, запустите **grub** для входа в оболочку *Grub*, затем введите

```
root (hd1,0)
setup (hd0)
quit
```

Первая команда определяет местонахождение файлов *Grub* – раздел, содержащий директорию **/boot/grub**; вторая записывает начальный код загрузчика в MBR; смысл третьей команды я сохраню в тайне.

При условии, что все запустилось без ошибок, после перезагрузки Вы увидите Ваше исходное меню во всей красе. Альтернатива – уберечь себя от проблем, полностью удалив XP. **НБ**

› Многие дистрибутивы, включая OpenSUSE, предлагают список совместимого оборудования на своих сайтах.

Нужна помощь!

› Для наилучшего ответа на ваш вопрос нам нужно знать как можно больше подробностей. Детально опишите конфигурацию системы. Если вы получили сообщение об ошибке, приведите текст сообщения и точно опишите вызвавшие его действия. Если у вас проблемы с оборудованием, то опишите его. Если Linux уже запущен, то выполните в root-терминале следующие команды и прикрепите к письму файл **system.txt**:

```
un[illegible] >system.txt
lspci >>system.txt
lspci -vv >>system.txt
```

› Пожалуйста, помните, что сотрудники журнала НЕ являются авторами или разработчиками Linux, любых пакетов или дистрибутивов. Зачастую люди, отвечающие за приложения, выкладывают большую часть информации на web-сайты. Попробуйте почитать документацию!

Мы стараемся ответить на все вопросы. Если вы не нашли ответ на свой, это, возможно, потому, что мы уже ответили на похожий вопрос.

Краткая справка про...

Расход памяти

Куда девается память? Вы замечали, что через пару часов работы на компьютере счетчик памяти свидетельствует: осталось лишь несколько свободных килобайт? Создается впечатление, что пора прикупить памяти; но на самом деле, сколько бы ее ни было, через пару часов после загрузки (а то и раньше) вся она окажется занятой.

В чем дело? Нет ли утечки памяти? Нет; просто ядро использует память очень эффективно. Расходовать память зря – это нехорошо; но еще хуже не использовать ее вообще. Вот ядро Linux и передает простаивающую память кэшу файловой системы и т.п., повышая производительность. Взгляните на количество буферов в выводе top и убедитесь: например,

```
Mem: 1028092k total, 1018704k used, 9388k
free, 49760k buffers
```

Из 1 ГБ ОЗУ на данном компьютере свободны всего 10 МБ. Половину используемой памяти занимают буферы. Они здорово сокращают время доступа, кэшируя данные в памяти и считывая сверх того, что вам нужно, в предвидении следующего обращения. Если память понадобится для чего-то поважнее, например, данных программы, она будет немедленно освобождена.

В памяти остаются не только данные файловой системы. Проследите за ее использованием при запуске и закрытии крупного приложения вроде *OpenOffice.org* или *Firefox*: Вы увидите, что объем свободной памяти уменьшается. Теперь снова запустите *Firefox* и отметьте, насколько быстрее программа загрузилась на этот раз: ядро сохранило данные в памяти для повторного использования.

То же относится к разделяемым библиотекам – вот почему бывает, что при смешанном запуске приложений KDE и Gnome система работает медленнее.

Большой вопрос Как включать и выключать удаленную машину?

10 Контроль над удаленным компьютером

У меня есть ПК в доме моего отца, и я не хочу держать его включенным 24 часа в сутки. Значит, мне нужен способ включать и выключать его из моего дома. На той машине запущен Windows XP, а у меня Slackware 12.0. Возможно ли это?

Eddygrinder, с форумов

Все, что Вам нужно для выключения компьютера – это запустить какое-нибудь ПО удаленного рабочего стола; затем Вы можете войти и выключить его из меню Пуск, как если бы Вы сидели напротив этой машины.

Если на компьютере стоит XP Home, хорошим выбором будет VNC (Virtual Network Computing). Для низкоскоростного интернет-соединения его реализацией является *TightVNC* (**www.tightvnc.com**). Установите программу на машину с Windows и настройте сервер на запуск. В этом случае Вам нужно открыть порты 5800 и 5900 на Вашем маршрутизаторе и брандмауэре. Если Вы используете KDE, для соединения с VNC на удаленном ПК можно взять KRDC, иначе установите на Вашу машину со Slackware *TightVNC* и используйте его для соединения с рабочим столом Windows.

Включение компьютера использует совершенно другую технологию, называемую Wake-on-LAN («побудка через LAN»). Когда компьютер выключен, но все еще соединен с источником питания, он слушает через свой сетевой интерфейс, не появится ли в сети «магический пакет» – специальный набор символов. Получив его, он включит сам себя. Это требует поддержки Wake-on-LAN в BIOS материнской платы. Большинство современных BIOS поддерживают его, но часто не активируют данную возможность по умолчанию; тогда найдите нужную опцию в меню настройки BIOS и включите ее. Если у Вас встроенная сетевая карта, больше ничего делать не придется, но в случае сетевой карты PCI потребуется специальный кабель для соединения разъемов Wake-on-LAN на ней и на материнской плате. Wake-on-LAN использует 9-й порт, поэтому обеспечьте пересылку пакетов с него на маршрутизаторе в Вашу сеть.

Все это заработает, если у Вас имеется постоянно включенный отдельный модем/маршрутизатор. Он также должен быть подключен к компьютеру через Ethernet, так как Wake-on-LAN работает только с адаптерами Ethernet. Наконец, Вам нужен аппаратный адрес Ethernet-адаптера на Windows-компьютере, который можно получить, запустив ipconfig в командной строке. Добыв эти данные, Вы можете запустить скрипт *wakeonlan* с **http://gsd.di.uminho.pt/jpo/software/wakeonlan** с помощью команды

```
wakeonlan -i [IP_адрес_сервера] [MAC адрес]
```

Например:

```
wakeonlan -i 123.124.125.126 00:0C:29:55:B0:C1
```

IP-адрес, используемый Вами с *wakeonlan* или *TightVNC*, должен быть Вашим внешним видимым адресом, а не внутренним сетевым адресом конкретной машины. Поскольку большинство провайдеров используют динамическое выделение адресов, Вам придется привлечь один из динамических сервисов DNS для привязки доменного имени к динамическому адресу; но иногда добрые провайдеры предоставляют статические IP-адреса. *Wakeonlan* должен справиться и с Вашим динамическим доменным именем машины вместо IP-адреса, а если нет, пропингуйте имя компьютера или запустите *dig*, чтобы получить IP-адрес. **НБ** LXF

Шаг за шагом: Удаленный контроль с VNC и Wake-on-LAN

1 Настройте VNC...

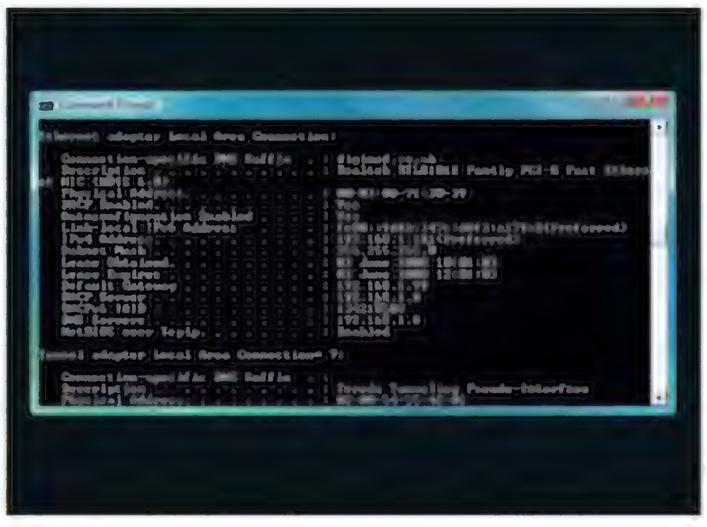
После установки TightVNC в Windows вам может понадобиться задать пароль, но для всего остального отлично подойдут настройки по умолчанию.

2 ...и маршрутизатор

Используйте web-интерфейс вашего маршрутизатора, чтобы открыть нужные порты для Windows-машины.

3 Спать пора!

Когда VNC запущен, вы можете получить доступ к рабочему столу Windows удаленно и выключить его обычным способом – нажав Пуск, конечно!



4 Узнайте свой адрес

Запустите ipconfig /all в командной строке и посмотрите MAC-адрес вашего компьютера с Windows; он вам понадобится для Wake-on-LAN.

5 Включите Wake-on-LAN

По умолчанию он обычно дезактивирован, поэтому зайдите в BIOS и включите его.

6 Подправьте свой IP

Если у вас динамический IP-адрес, вам необходимо использовать сервис динамического DNS для соединения с удаленной машиной.

LXF DVD109

Двусторонний диск с самыми свежими дистрибутивами!

Помимо новых релизов OpenSUSE и Linux Mint, у нас на DVD есть еще три версии Linux: 64 Studio – мультимедийный дистрибутив, разработанный для «создателей цифрового контента»; Mandriva One 2008 Spring – для установки на Eee PC; и Puppy Linux 4.00 – дистрибутив-крошка, способный реанимировать старые машины. Если у вас есть старый компьютер, в котором еще теплится жизнь, подарите ему немного щенячьей любви [puppy – англ. «щенок»].

Майк Сондерс
Редактор диска
mike.saunders@futurenet.co.uk

Дистрибутив Linux

OpenSUSE 11.0

Эволюция, не революция – вот основная философия разработки OpenSUSE 11.0, и, как следствие, дистрибутив не назовешь под завязку напичканным новыми супер-пупер-функциями. Данный релиз в большей степени посвящен отладке и настройке – кодовая база OpenSUSE 10.3 плюс исправление мелких шероховатостей и более гладкая и ровная работа во всем. Постоянным пользователям SUSE стоит обновиться, а если вы раньше не встречались с дистрибутивом, созданным Novell при поддержке сообщества, рекомендуем познакомиться с ним. На момент написания данного материала OpenSUSE занимает вторую строчку в хит-параде популярности DistroWatch.com, а поскольку история создания этого дистрибутива уходит корнями в 90-е годы, у него множество верных фанатов.

Но, тем не менее, в 11.0 есть достаточно новых функций, которые стоит изучить, в том числе:

» **KDE 4** Это первый релиз OpenSUSE, в котором KDE 4.0 поставляется по умолчанию. Рабочий стол отличается обычным для Novell блеском, и это – отличная возможность испробовать систему виджетов Plasma (подробнее о них чуть позже).

» **Инсталлятор** Благодаря таблицам стилей Qt 4.4, интерфейс стал весьма и весьма привлекательным.

» **Управление пакетами** Здесь было сделано множество исправлений и усовершенствований, так что теперь установка новых программ происходит намного быстрее.

» **Compiz Fusion** Теперь интересные эффекты рабочего стола активируются по умолчанию (при условии, что ваша видеокарта их поддерживает). Ура!

Более подробный материал по релизу 11.0 составляет тему этого номера **LXF**. Без сомнения, вам не терпится познакомиться с ним, поэтому мы рады предложить вам версию Live на **LXF**DVD – изучайте дистрибутив, не отказываясь от привычной системы. Если же вы

Шаг за шагом: Устанавливаем OpenSUSE 11.0

1 Загрузка

Запустите ваш ПК с **LXF**DVD, и когда появится этот экран меню, нажмите Enter. Если при старте OpenSUSE возникнут проблемы, перезагрузитесь и выберите опцию Failsafe.

2 Рабочий стол

Через пару минут перед вами появится рабочий стол OpenSUSE Live – он работает прямо с DVD. Нажмите на X справа снизу от этого информационного окна, чтобы избавиться от него.

5 Настройка

Если вы выберете Create Partition Setup для разбиения диска вручную, создайте корневой раздел root в формате ext3 размером не менее 5 ГБ и раздел подкачки для виртуальной памяти размером 512 МБ.

6 Учетная запись

После этого вас попросят создать постоянную учетную запись пользователя (не администратора). Не забудьте свой пароль! Здесь также можно настроить автоматический вход в систему.

захотите оставить OpenSUSE навсегда, можно сделать и полную установку. На DVD представлена 32-битная версия, но она будет прекрасно работать и на x86-64. Для инсталляции вам потребуется не менее 256 МБ ОЗУ и 5 ГБ места на жестком диске.

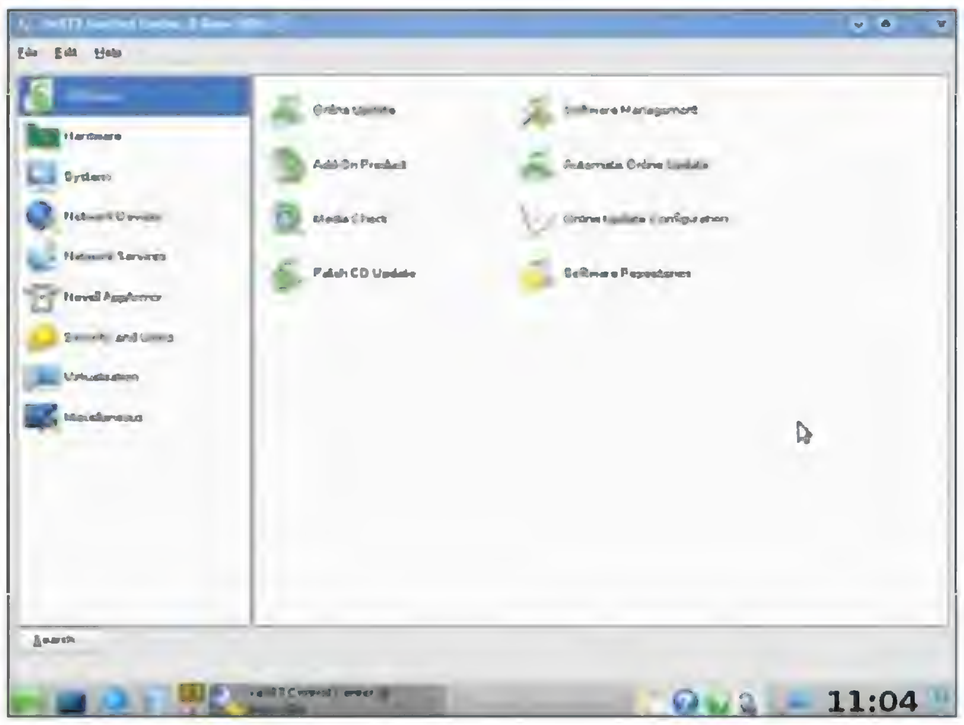
OpenSUSE 11.0 загружается прямо с **LXF**DVD: просто вставьте диск в дисковод и нажмите Enter, когда появится меню. Если ваш ПК настроен на запуск с жесткого диска, а не с DVD, вам надо будет изменить настройки BIOS, как описано на стр. XX. Примерно через минуту вы увидите рабочий стол KDE 4, где вы сможете опробовать виджеты Plasma – кликните по желтой кривой в правом верхнем углу экрана, чтобы открыть окно менеджера. После этого вы сможете добавлять на ваш рабочий стол мини-апплеты, такие как часы, калькулятор или RSS-агрегатор (если у вас есть соединение с Интернетом).

Гений настройки

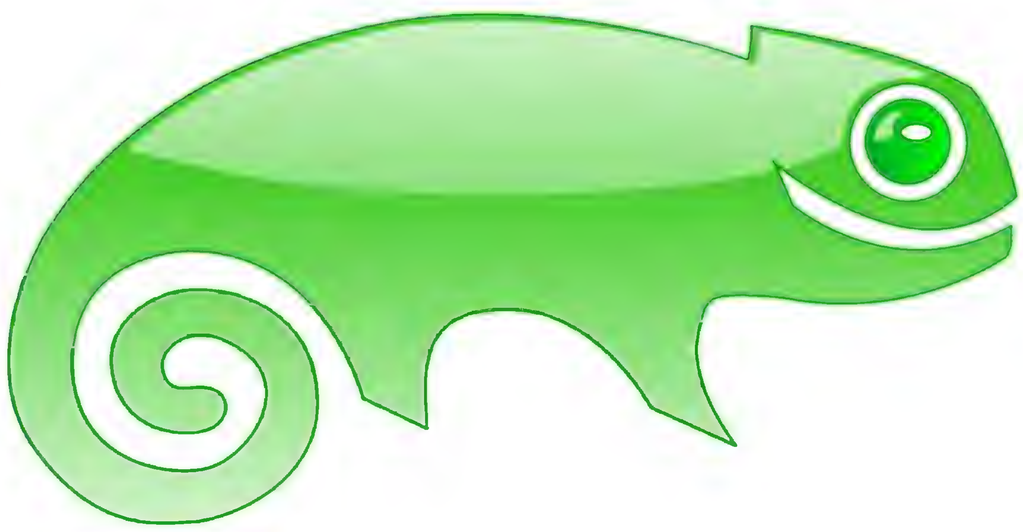
Yast, всеобъемлющий мега-конфигуратор OpenSUSE, имеет своих сторонников и противников, но это, несомненно, бесконечно мощная программа. Через Yast вы можете управлять практически любым аспектом своей системы, от пакетов программ до сетевых сервисов и учетных записей. Yast, как и все приложения, включенные в поставку OpenSUSE, можно запустить, нажав на зеленую кнопку с гекконом в левом нижнем углу экрана. Найдите во всплывающем меню раздел Computer, а затем нажмите Yast на самом верху.

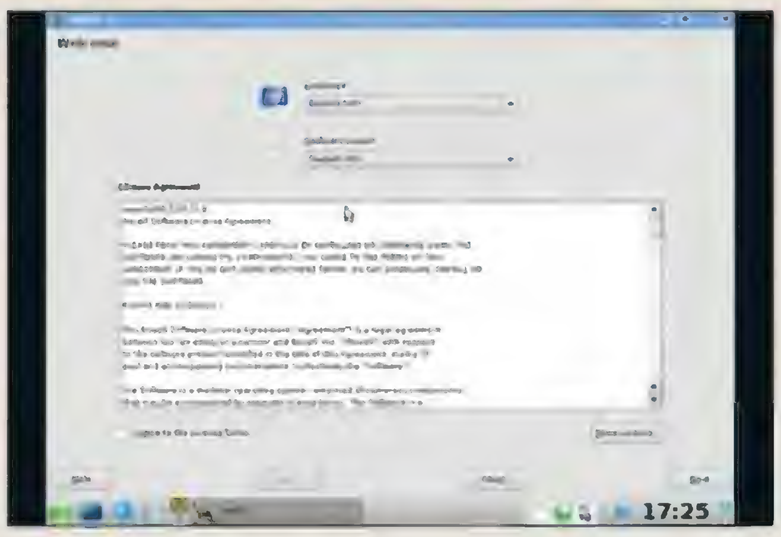
Во время установки вам предложат разбить ваш жесткий диск на разделы. Если вы – матерый линуксоид, у вас не будет с этим проблем – просто удалите существующий дистрибутив или выделите дополнительное место (5 ГБ – минимум; рекомендуется 10+ ГБ). Если же вы – новичок, сообщаем, что вам понадобится отвести на жестком диске пространство для OpenSUSE. Программа разбиения диска предложит вам несколько опций, позволяя создать двойную загрузку с Windows, если вы хотите, чтобы у вас были установлены обе операционные системы. Если вы не планируете оставлять на своем ПК никакую другую ОС, можете отдать во владение OpenSUSE весь винчестер целиком – это самый простой вариант.

По-хорошему, вы не должны встретить никаких чудовищ ни во время установки, ни во время запуска OpenSUSE 11.0. В маловероятном случае появления проблем, или если у вас просто возникнут вопросы по дистрибутиву, помощь всегда рядом. Загляните на Линуксфорум (www.linuxforum.ru), где вам предложат дружескую поддержку другие читатели **LXF** и не только – не забудьте указать всю необходимую информацию о возникшей неполадке, в том числе, параметры своего компьютера. Еще один полезный ресурс, где можно получить содействие – http://forums.opensuse.org, а общую информацию о дистрибутиве можно найти на сайте http://en.opensuse.org. Как говорит классический лозунг SUSE, большого вам удовольствия!



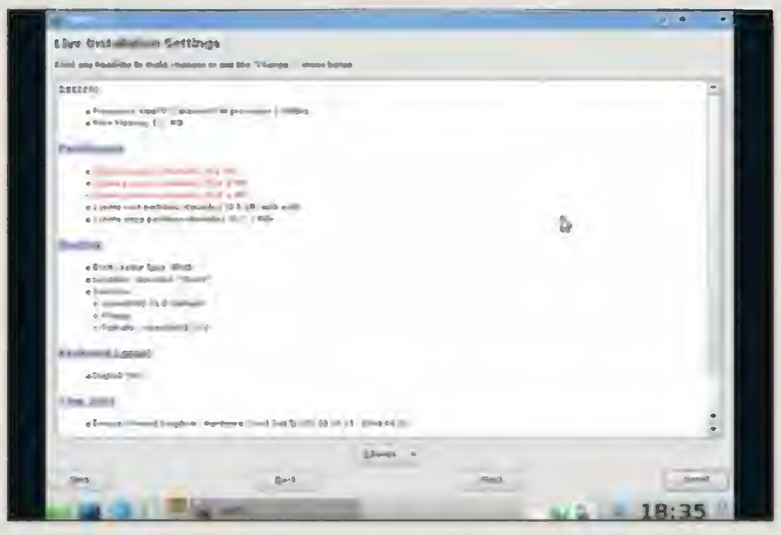
› ***Yast*** **– это конфигуратор-комбайн, предоставляющий управление пакетами, инструменты настройки оборудования и прочие функции.**





3 Инсталляция
Готовы к установке? Нажмите на значок Install на рабочем столе. Через несколько мгновений появится окно программы установки *Yast*, предлагая настроить язык и раскладку клавиатуры.

4 Разделы
После выбора часового пояса вам будет предложено разбить ваш жесткий диск на разделы – это можно делать автоматически или вручную. См. текст выше, где описан этот процесс.



7 Подтверждение
OpenSUSE покажет предполагаемые параметры установки, и это – последний шанс что-то в них изменить (просто щелкните по синей ссылке). Если же вы всем довольны, нажмите кнопку Install.

8 Отдохните
Начнется копирование файлов OpenSUSE – на более старых машинах это может занять около часа. Как только файлы скопируются, перезагрузитесь, извлеките DVD и ныряйте в новую установку OpenSUSE!

Дистрибутив Linux

Linux Mint 5

Дистрибутивы приходят и уходят, подобно временам года, но Linux Mint, похоже, остается. Появившийся изначально как мелкомасштабное ответвление Ubuntu с дополнительными мультимедиакодеками, Mint за прошедшие 12 месяцев рос взрывными темпами и занял более чем почетное четвертое место в хит-параде **DistroWatch.com**, частично благодаря своей начинке от Ubuntu и успеху других дистрибутивов – потомков Ubuntu, но не меньшую роль в этом достижении сыграло то, что проект с лазерной точностью фокусируется на внешней привлекательности и эргономичности.

И хотя Mint 5 в принципе являет собой слегка подправленную версию Ubuntu 8.04, с обычными приложениями рабочего стола типа Firefox и OpenOffice.org, в некотором отношении он выглядит как совершенно отдельный дистрибутив. У него тщательнейшим образом отлаженный рабочий стол Gnome плюс подборка специфических инструментов Mint для управления программами. Например, если вы нажмете на кнопку «Elyssa» (это кодовое название дистрибутива) в нижней левой части экрана, перед вами появится меню Mint, где перечислены установленные программы. При желании можно щелкнуть правой кнопкой по названию программы и удалить ее.

Самый мятный

[Mint означает мята, – прим. пер.] Mint 5 загружается прямо с LXFDVD: запустите ваш ПК, вставив диск в привод, и выберите в меню загрузки Linux-Mint-5. (Если при загрузке возникнут проблемы, попробуйте опцию Safe Graphics.) Через пару минут вы увидите рабочий стол, позволяющий познакомиться с дистрибутивом, не затрагивая жесткий диск. Если же вы решите установить его, отчего он заработает намного быстрее, дважды щелкните по значку Install на рабочем столе. Процесс инсталляции такой же, как в Ubuntu 8.04, и системные требования не отличаются: 700-МГц 32-битный или x86-64 процессор, 384 МБ ОЗУ и 8 ГБ свободного места на жестком диске. Если у вас сейчас Windows, Mint позволит создать двойную загрузку на стадии выделения разделов – то есть при загрузке будет появляться меню, позволяющее выбрать между Linux и Windows. Прежде чем начать установку, обязательно сделайте резервные копии всех важных данных Windows на случай, если что-то пойдет не так! Наслаждайтесь дистрибутивом, а если вам понадобится помощь, вы найдете массу полезной информации на **www.linuxmint.com**.

Очень ВАЖНО!

Перед тем, как вставить DVD в дисковод, пожалуйста, убедитесь, что вы прочитали, поняли и согласились с нижеследующим:

Диски *Linux Format* DVD тщательно проверяются на предмет отсутствия на них всех известных вирусов. Тем не менее, мы рекомендуем вам всегда проверять любые новые программы надежным и современным антивирусом.

Хотя процесс отбора, тестирования и установки программ на DVD проводится со всем тщанием, редакция *Linux Format* не несет никакой ответственности за повреждение и/или утрату данных или системы, могущее произойти при использовании данного диска, программ или данных на нем. Настоятельно рекомендуем вам создавать своевременные и надежные резервные копии всех важных файлов.

Чтобы узнать об условиях использования, просим вас прочесть лицензии.

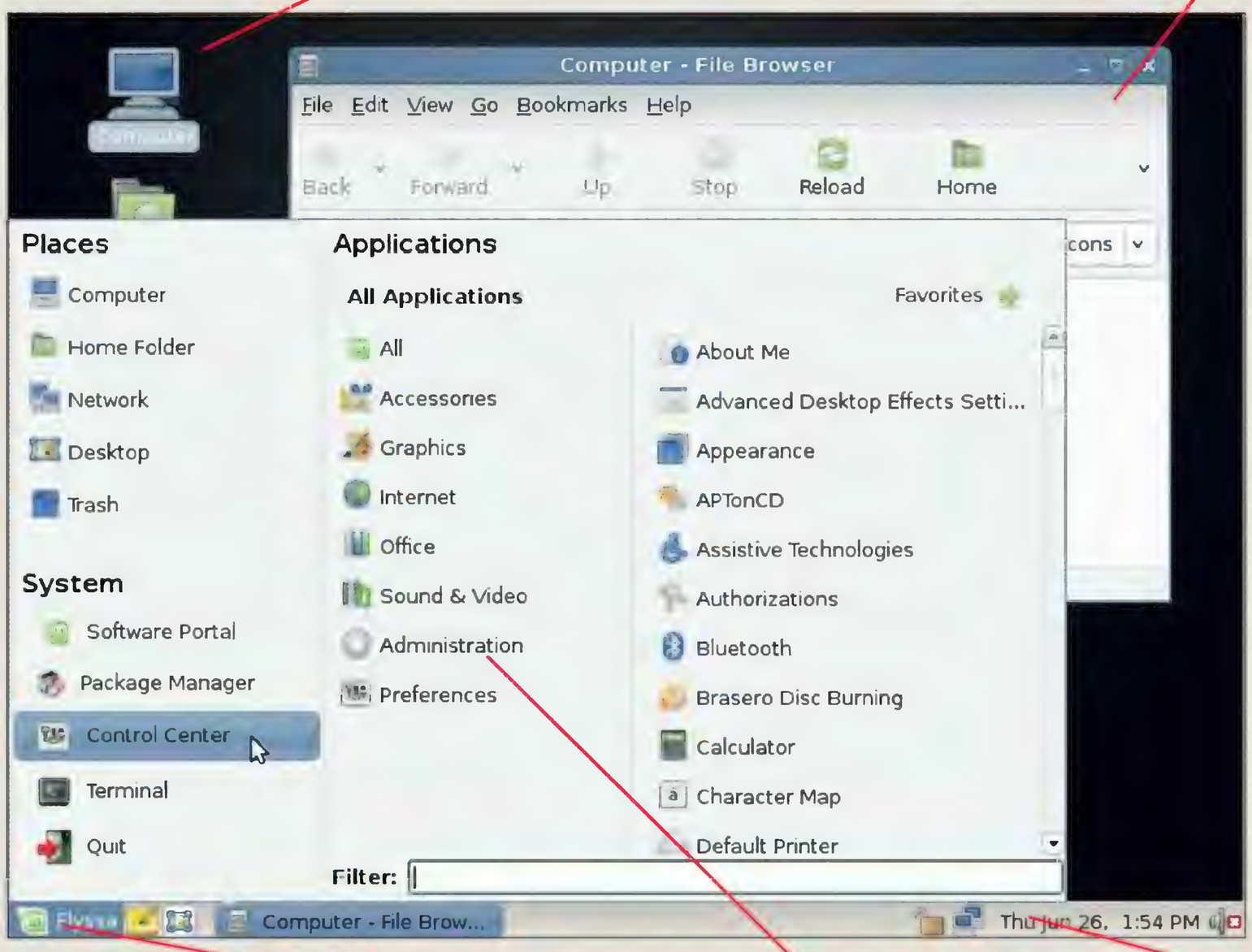
Бракованные диски

В маловероятном случае обнаружения бракованного диска *Linux Format*, просим связаться с нашей группой поддержки по адресу **disks@linuxformat.ru** для получения содействия.

Исследуем Mint 5

Рабочий стол
Отсюда вы можете получить доступ к своей домашней директории; здесь же можно хранить временные файлы.

Файлы
Обозреватель *Nautilus* – стандартный способ навигации по файловой системе.



Меню
Нажмите на кнопку Elyssa снизу слева, чтобы вывести список …

Приложения
… установленных программ. Откройте Control Centre [Центр Управления] для изменения настроек или Package Manager [Менеджер Пакетов], чтобы добавить новые приложения.

Системный лоток
В этой области на панели задач отображаются значки работающих в фоновом режиме программ и сетевых соединений.

Сентябрь 2008

LXFDVD109

Дистрибутивы для опытных пользователей

» Slackware 12.1 Самая старая из ныне существующих разновидностей Linux, поддерживаемая Патриком Фолькердингом.

» Arch Linux 2008.06 Чрезвычайно гибкий дистрибутив, унаследовавший идеи Slackware и CRUX.

openSUSE 11.0

Блестящий, быстрый, красивый Linux

» Запускается в режиме Live: Установка не требуется.

» KDE 4 – получите рабочий стол будущего уже сегодня.

» Быстрое и простое управление пакетами.

Также на DVD: Linux Mint 5.0

ВНИМАТЕЛЬНО ПРОЧТИТЕ ЭТО ПЕРЕД ТЕМ, КАК ИСПОЛЬЗОВАТЬ ДВД-ДИСК !

Информация о диске

ЧТО-ТО ПОТЕРЯЛИ?

Часто случается, что новые программы зависят от других программных продуктов, которые могут не входить в текущую версию вашего дистрибутива Linux.

Мы стараемся предоставить вам как можно больше важных вспомогательных файлов. В большинстве случаев, последние версии библиотек и другие пакеты мы включаем в каталог «Essentials» (Главное) на прилагаемом диске. Поэтому, если в вашей системе возникли проблемы с зависимостями, следует заглянуть именно туда.

ФОРМАТЫ ПАКЕТОВ

Мы стараемся включать как можно больше различных типов установочных пакетов: RPM, Deb или любые другие. Просим вас принять во внимание, что мы ограничены свободным пространством и доступными бинарными выпусками программ. По возможности, мы будем включать исходные тексты для любого пакета, чтобы вы смогли собрать его самостоятельно.

ДОКУМЕНТАЦИЯ

На диске вы сможете найти всю необходимую информацию о том, как устанавливать и использовать некоторые программы. Пожалуйста, не забывайте, что большинство программ поставляются вместе со своей документацией, поэтому дополнительные материалы и файлы находятся в соответствующих директориях.

ЧТО ЭТО ЗА ФАЙЛЫ?

Если вы новичок в Linux, вас может смутить изобилие различных файлов и расширений. Так как мы стараемся собрать как можно больше вариантов пакетов для обеспечения совместимости, в одном каталоге часто находятся два или три файла для различных версий Linux, различных архитектур, исходные тексты и откомпилированные пакеты. Чтобы определить, какой именно файл вам нужен, необходимо обратить внимание на его имя или расширение:

имя_программы-1.0.1.i386.rpm – вероятно, это бинарный пакет RPM, предназначенный для работы на системах x86;

имя_программы-1.0.1.i386.deb – такой же пакет, но уже для Debian;

имя_программы-1.0.1.tar.gz – обычно это исходный код;

имя_программы-1.0.1.tgz – тот же файл, что и выше по списку: «tgz» - это сокращение от «tar.gz»;

имя_программы-1.0.1.tar.bz2 – тот же файл, но сжатый bzip2 вместо обычного gzip;

имя_программы-1.0.1.src.rpm – также исходный код, но поставляемый как RPM-пакет для упрощения процесса установки;

имя_программы-1.0.1.i386.FC4.RPM – бинарный пакет RPM для x86, предназначенный специально для операционной системы Fedora Core 4;

имя_программы-1.0.1.ppc.Suse9.rpm – бинарный пакет RPM, предназначенный специально для операционной системы SUSE 9.x PPC;

имя_программы-devel-1.0.1.i386.rpm – версия для разработчиков.

Если диск не читается...

Это маловероятно, но если все же прилагаемый к журналу диск поврежден, пожалуйста, свяжитесь с нашей службой поддержки по электронной почте: **disks@linuxformat.ru**

Сторона 1

Рабочий стол
Banshee - аудиоплеер и каталогизатор
Blender - 3d-редактор
DigiKam - просмотрщик и каталогизатор фотографий
Euler - математический пакет
FreeMat - ещё одна альтернатива Matlab
F-Spot - программа для управления фотоколлекцией
GNU Octave - высокоуровневый язык численных расчётов
OpenOffice.org - офисный пакет
Sage - инструмент для изучения математики
Sciab - платформа для численных расчётов
Scribus - программа для профессиональной верстки

Разработка
Blue - новый динамический язык программирования
CMake - кроссплатформенная система сборки
Ctalk - объектно-ориентированные функции для C
dlib - библиотека для C++
DrScheme - графическое окружение для Scheme
GCC - коллекция компиляторов GNU
Nemiver - графический отладчик
PyQt - набор привязок Qt к Python
Qt - инструментарий для разработчиков
R - язык статистической обработки данных
Seed7 - язык программирования

Дистрибутивы
64 Studio - дистрибутив для работы с аудиоданными
Linux Mint - вариация Ubuntu Linux
Mandriva One - живой дистрибутив на базе Mandriva
OpenSUSE 11.0 - новая версия популярного дистрибутива
Puppy Linux - компактный дистрибутив для слабых машин

Система
cfv - утилита для создания верификационных файлов
jpwgen - генератор паролей на Java
MadWifi - драйвер для беспроводных карт
TimeVault - интерфейс для создания снимков директорий
xorriso - программа для работы с ISO 9660

Игры
FreeCol - свободная версия Colonization
Hedgewars - бои ёжиков
Regame - необычная аркада
RetroVaders - аналог Space Invaders

Интернет
Firefox - популярный веб-браузер
Flock - социальный браузер
Galaxium - клиент сетей быстрых сообщений
KTrafficAnalyzer - анализатор трафика
Twitim - XMPP/Jabber-клиент для Twitter

HotPicks
BG-Bootdisk - загрузочный диск
BlueProximity - программа для защиты вашего компьютера
EiffelStudio - среда разработки для Eiffel
nPush - логическая игра
PaperBox - просмотрщик документов
Psychosynth - модульный синтезатор
TuxGuitar - программа для гитаристов
Wormux - клон Worms
yaGTD - утилита для управления списком дел
Zile - маленький клон редактора Emacs

Справка
Ответы на часто задаваемые вопросы
Новичку в Linux
RUTE - книга по администрированию Linux

Сервер
phpMyAdmin - инструмент администрирования MySQL
TangoCMS - система управления контентом на PHP
WikyBlog - приложение для совместной работы

Сторона 2

Slackware Linux 12.1

Arch Linux 2008.06

Комментарии? Присылайте ваши мысли и предложения по электронной почте: info@linuxformat.ru
Пожалуйста, ознакомьтесь с опубликованной в журнале инструкцией перед использованием данного диска.
Настоящий диск тщательно тестировался и проверялся на всех стадиях производства, однако, как и в случае с любым новым ПО, мы рекомендуем вам использовать антивирусный сканер. Мы также рекомендуем всегда иметь под рукой актуальную резервную копию данных вашего жесткого диска. К сожалению, редакция Linux Format не может принимать на себя ответственность за любые повреждения, разрушения или иные убытки, которые может повлечь за собой использование этого DVD, представленных на нем программ или данных. Перед тем, как устанавливать какое-либо ПО на компьютер, подключенный к сети, проконсультируйтесь с сетевым администратором.

Дефектные диски. В маловероятном случае обнаружения дефектов на данном диске, пожалуйста, обращайтесь по адресу: disks@linuxformat.ru

Тираж изготовлен ООО «Фортмедиа», Россия, Санкт-Петербург, 196006 ул. Цветочная д. 7, тел. +7 (812) 388 8200. Лицензия МПТР России ПАФ № 77-225

Поставляется вместе с журналом LINUXFORMAT номер 9 (109) сентябрь 2008

СОЗДАНИЕ УСТАНОВОЧНЫХ ДИСКОВ ПРИ ПОМОЩИ CDRECORD

Самый быстрый способ записать ISO-образ на чистую матрицу – это *cdrecord*. Для всех перечисленных ниже действий потребуются права root. Для начала определите путь к вашему устройству для записи дисков. Наберите следующую команду:

```
cdrecord -scanbus
```

После этого на экране терминала должен отобразиться список устройств, подключенных к вашей системе. SCSI-адрес каждого устройства представляет собой три числа в левой колонке, например, 0,3,0. Теперь вы можете с легкостью записать образ на диск:

```
cdrecord dev=0,3,0 -v /путь к образу/image.iso
```

Чтобы упростить дальнейшее использование *cdrecord*, сохраните некоторые настройки в файле **/etc/default/cdrecord**. Добавьте по одной строке для каждого устройства записи (вероятно, в вашей системе присутствует всего одно такое устройство):

```
Plextor= 0,3,0 12 16M
```

Первое слово в этой строке – это метка, затем, после адреса SCSI-устройства вы должны указать скорость и размер буфера. Теперь вы можете заменить SCSI-адрес в командной строке на выбранную вами метку. Все будет еще проще, если вы добавите следующее:

```
CDR_DEVICE=Plextor
```

Все, что вам теперь нужно для записи ISO-образа – это набрать команду

```
cdrecord -v /path/to/image.iso
```

Если вы не из числа любителей командной строки, в таком случае вам придет на помощь утилита *gcombust*. Запустите ее из-под root, выберите вкладку Burn и ISO 9660 Image в верхней части окна. Введите путь к образу, который вы хотите записать на диск, и смело нажимайте на Combust!. Пока ваш образ пишется на диск, можете выпить чашечку кофе.

Другая ОС?

Вам не обязательно использовать Linux для записи компакт-диска. Все необходимые файлы уже включены в ISO-образ. Программы вроде *cdrecord* просто переносят данные на чистую матрицу. Если у вас нет устройства для записи дисков, можно найти того, у кого оно есть, и записать диск на его компьютере. На нем может стоять Windows, Mac OS X, AmigaOS, или любая другая ОС.

Нет устройства для записи дисков?

А что, если у вас нет устройства, с помощью которого можно было записать образ на диск? Вы знаете кого-либо с таким устройством? Вам не обязательно использовать Linux для записи дисков, подойдет любая операционная система, способная распознать пишущий привод (см. выше).

Некоторые дистрибутивы умеют монтировать образы дисков и выполнять сетевую установку или даже установку с раздела жесткого диска. Конкретные методы, конечно, зависят от дистрибутива. За дополнительной информацией обращайтесь на web-сайт его разработчика.

Новое ПО

Кодирование и рабочий стол

Помимо дистрибутивов, на DVD более 50 полезных приложений и утилит. В разделе Рабочий стол мы предлагаем *Banshee 1.2*, свежий полный релиз очень мощного и отлаженного медиа-плейера. Написанный на C# (и по этой причине использующий среду Mono), *Banshee* так и сыплет функциями из каждой строки кода: в нем имеются отдельные музыкальные и видео библиотеки, он может связываться с iPod, поддерживает интеграцию с Last.fm, позволяет подписываться на подкасты и умеет даже выполнять риппинг и прожиг CD благодаря Brasero. Предусмотрен «мини режим» (mini mode) – он сокращает интерфейс, если вы захотите, чтобы плейер работал где-то в уголке экрана.

Banshee требует установки Mono 1.2.2 или более новой версии – она должна найтись в менеджере пакетов вашего дистрибутива. Скопируйте **banshee-1-1.2.1.tar.bz2** в домашнюю директорию, откройте терминал и введите:

```
tar xfvj banshee-1-1.2.1.tar.bz2
cd banshee-1-1.2.1/
./configure && make
```

Затем переключитесь на пользователя root (в разных дистрибутивах это выполняется по-разному, но всегда сгодится **su** или **sudo bash**) и введите **make install**. Если все пройдет нормально, для запуска приложения вы сможете просто ввести **banshee** в командной строке. Еще можно создать пункт в меню или ярлык быстрого запуска приложения (способы опять-таки варьируются от дистрибутива к дистрибутиву).

Также в разделе Рабочий стол имеется *OpenOffice.org 2.4.1* Pro Это мелкое исправление версии 2.4, так что стоит установить его, если у вас были трудности с 2.4.0. Мы попытались уместить на нашем DVD двоичные сборки для многих популярных дистрибутивов; если вашего нет в этом списке – возьмите универсальный архив, **OOo_2.4.1-1_LinuxIntel_ru_infra.tar.gz** Будет лучше заранее удалить старую версию, а после установки OOo 2.4.1 вы найдете его в каталоге **/opt**. Более подробная информация по работе в командной строке и установке программ – в разделе **Справка/Новичку в Linux** нашего DVD.

Это… Firefox 3.0!

В разделе Интернет вас ждет новая версия Firefox. Благодаря огромной проделанной работе, релиз 3.0 оказался весьма впечатляющим, с ускоренным движком JavaScript и множеством исправленных утечек памяти. Если вам нравится *Firefox 2*, но вы считаете его малость неповоротливым для сайтов Web 2.0 типа Google Docs, испытайте этот новый релиз – он намного быстрее. В нем также имеется новая адресная строка, способная осуществлять поиск не только по ранее посещенным URL (как в 2.0), но также и по названиям web-страниц. Чтобы запустить его, скопируйте **firefox-3.0.1.tar.bz2** с DVD в домашнюю директорию, откройте терминал, и введите:

```
tar xfvj firefox-3.0.1.tar.bz2
firefox/firefox
```

Таким образом вы распакуете архив, создав новый каталог, и затем запустите из него программу **firefox**.

Те, кто в настроении заняться кодированием, найдут массу интересных инструментов в разделе Разработка. Вы уже знакомы с *GCC*, *Qt* и *PyQt*, так что мы здесь фокусируем внимание на других. Blue – новый язык программирования с минималистским синтаксисом, простой в восприятии, вне зависимости от того, чем вы пользовались раньше – такова была идея разработки. Он компилируется в байт-код и выполняется в виртуальной машине. Seed7 – еще один новый язык; авторы сравнивают его с C++, Java и Ada. Его свойства: объектно-ориентированность, работа с исключениями и перегрузка функций/операторов, а еще Seed7 конвертируется в код C, который затем компилируется в двоичный, как обычно. И, наконец, предлагаем вам DrScheme, полную реализацию языка программирования Scheme с мощной IDE. LXF

› Мало что может сравниться с *Banshee* по эффективности управления видео, музыкой и коллекцией подскастов.

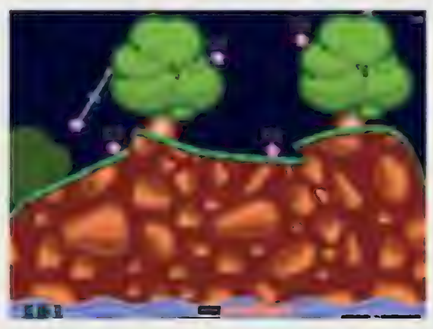
› Seed7 не ограничивается простыми программами командной строки – вы можете писать на нем игры вроде этой, Castle.

Развлечения

Пять лучших игр

В разделе Игры вы найдете пять отличных пожирателей вашего времени, которые помогут скоротать перерыв на обед. Первый – *FreeCol*, пошаговая стратегия, основанная на классической *Colonization* Сида Мейера [Sid Meier] из 1994. FreeCol написана на Java, и вам понадобится установить JRE (Java Runtime Environment) – вы найдете ее в менеджере пакетов своего дистрибутива. Скопируйте **freecol-0.7.4-installer.jar** в домашнюю директорию, откройте окно командной строки и введите **java -jar freecol-0.7.4-installer.jar**, чтобы запустить игру. Более подробную информацию вы найдете в **manual.pdf** на **LXF**DVD.

Далее идет *Hedgewars*, блестящая многопользовательская игра в стиле *Worms* – но с ежиками. Даже незнакомые с *Worms* легко поймут простые правила: вы по очереди управляете персонажами в двухмерном мире, пытаясь прицельным огнем уничтожить противника. Навык вырабатывается из умения оценить угол и силу выстрела – вам понадобится зоркий глаз и отличное чувство расстояния, чтобы размазать своих врагов.



› *Ежевойны*: ежи с ежами единоборствуют, ежатся, ершатся, кто ежистее решают.

И еще – *Magic Maze*, игра в традициях старой школы с видом сверху; в ней вам предстоит направлять волшебника по лабиринту, сражаясь с монстрами и отыскивая ключи, чтобы выйти наружу.

А вот *Regame*, в свою очередь, потребует от вас совершенно нового уровня экологического сознания: это игра освобождения от отходов. Вам предстоит совершить путешествие во времени из будущего, спасая планету Земля, увертываясь по пути от летающих бананов. Э-э, ну, ладно…И наконец, чтобы вы в полной мере ощутили ностальгию, предлагаем вам *RetroVaders*, превосходную возможность вернуться к видеоиграм прошлого – с открытым кодом. LXF

» Лучшие новинки открытого ПО на планете

LXF HotPicks

Ричард Смедли
Активист сообщества FOSS и давний сотрудник LXF, Ричард бороздит глубины с умницей-дельфином в поисках сокровищ Hotpicks.

В ЭТОТ РАЗ ТОЛЬКО ДЛЯ ВАС: » Psychosynth » BlueProximity » Paperbox » EiffelStudio » Wormux » NPush » TuxGuitar » BG-Tiny Linux Bootdisk » YAGTD » Zile

Модульный синтезатор

Psychosynth

Версия 1.4.15 **Сайт** www.psychosynth.com

Забава! Вот мощнейшая мотивация разработки. Ричард Столлмен писал как-то, что истинное хакерство сродни игре. *Psychosynth* полностью соответствует этому определению.

Основанный на Reactable инструмент подразумевает активное сотрудничество, через перемещение физических блоков по виртуальному столу, для контроля за создаваемым звуком. *Psychosynth* предоставляет вам виртуальный рабочий стол и набор стандартных «аудиоблоков». Начните с одного-двух звуковых генераторов и нескольких сэмплов. К коллекции OGG-файлов программы можно добавлять новые – если у вас есть собственные наработки, поделитесь с проектом, от этого выиграют все.

Итак, повеселимся: любые звуки генераторов и сэмплов можно беспрепятственно менять с помощью органов управления, фильтров, эффектов и микшеров. Звуки можно объединять в любых сочетаниях – в точности как на синтезаторе 70-х, только гораздо более свободно и разнообразно, и вдобавок с немедленной визуальной отдачей.

Музыка на столе

Разместив музыкальные блоки, можно вертеть стол в любой плоскости, увеличивать его или уменьшать для лучшего обзора. Выделив генератор или эффект, можно менять его частоту, громкость и другие характеристики, подправив соответствующие управляющие установки.

> «*Psychosynth* дает вам виртуальный стол и набор готовых мелодий.»

Руководство пользователя очень полезное, но не всем – пока оно доступно только на испанском; ау, переводчики! Популярность

› Руководство пользователя пока доступно лишь на испанском, но визуальные метафоры *Psychosynth* понятны без слов.

проекта Reactable постепенно растёт: посетите http://reactable.iua.upf.edu/?related и убедитесь, что он нравится не только нам.

Есть версия *Psychosynth* для командной строки, она будет весьма кстати для готового звука, когда эксперименты окончены: ведь *Psychosynth* таки отъедает циклы процессора. Помните об этом, если ваш CPU одноядерный. У нас разок случилось зависание, но для столь ранней версии это простительно. В остальном же это отличное развлечение и превосходная альтернатива загромождению реальной мебели.

Шаг за шагом: Давайте пошумим!

» Розовый шум
Щёлкните круглый значок в нижнем левом углу, чтобы открыть окно выбора объектов. Выберем генератор «розового» шума.

» Звуковой стол
Начнём с пары звуковых генераторов, затем добавим несколько фильтров и послушаем, что получится.

» Какофония под контролем
Щелчком на фильтре можно вращать внешнее кольцо, меняя уровень, частоту и прочие параметры звука.

Безопасность

BlueProximity

Версия 12 **Сайт** http://blueproximity.sourceforge.net

Есть немало поводов оберегать свою работу от чужих взоров: соблюдение конфиденциальности клиента; наличие соседа-конкурента; работа с ноутбуком среди посторонних; наконец, желание скрыть от начальства просмотр сайта **LXF** в рабочее время. Отрываясь от клавиатуры, лучше не оставлять дисплей без присмотра.

Если вас куда-то отозвали, можно активировать заставку, уйти на виртуальную консоль или просто сменить рабочее окно – слабая, но защита. Заставка с паролем повышает уровень безопасности, но всегда ли вы вспомните о ней, отлучаясь на несколько секунд?

В *BlueProximity* Ларс Фридрихс [Lars Friedrichs] нашёл удобное решение этой проблемы умственных тружеников. Приложение постоянно отслеживает дистанцию между компьютером и мобильным телефоном, оснащённым функцией Bluetooth. Стоит отойти от ПК с мобильником в кармане, и экран заблокируется. Вы вернетесь, и блокировка снимется автоматически – даже без ввода пароля.

Безопасность в кармане

Блокировка экрана – не единственная возможность: как реакцию на изменение расстояния можно задать любую команду. Можно назначить время, после которого экран запирается при вашей отлучке или разблокируется по возвращении. Настраивается и расстояние, при котором регистрируются уход и приход, причём можно связать с компьютером не одно Bluetooth-устройство.

Новой программе легко придумать массу применений. Если ПК стоит рядом с кухней, пусть она сосчитает, сколько раз вы варили кофе, а затем при помощи скрипта ежедневно рассылает по почте данные о вашем потреблении кофеина. Сказать по правде, про некоторые вещи лучше бы и не знать.

› Установите необходимое расстояние. Теперь отойдите с мобильником чуть подальше – экран заблокируется.

> «*BlueProximity* оценивает расстояние между ПК и вашим мобильником.»

Браузер на основе тегов

Paperbox

Версия 0.2 **Сайт** http://live.gnome.org/paperbox

PaperBox – удобный браузер документов для рабочего стола Gnome, применяющий тэговые и специфичные для содержимого данные при просмотре электронных книг и прочих документов. *PaperBox* использует данные *Tracker*, мощного Gnome-инструмента индексирования и управления базами метаданных.

Способность *Tracker* сохранять миниатюры файлов облегчает визуальное распознавание документов различных типов (эскизы отображаются рядом с тэговой информацией и названием файла). Тэги выводятся в облаке с правой стороны окна, а повыше помещается перечень категорий.

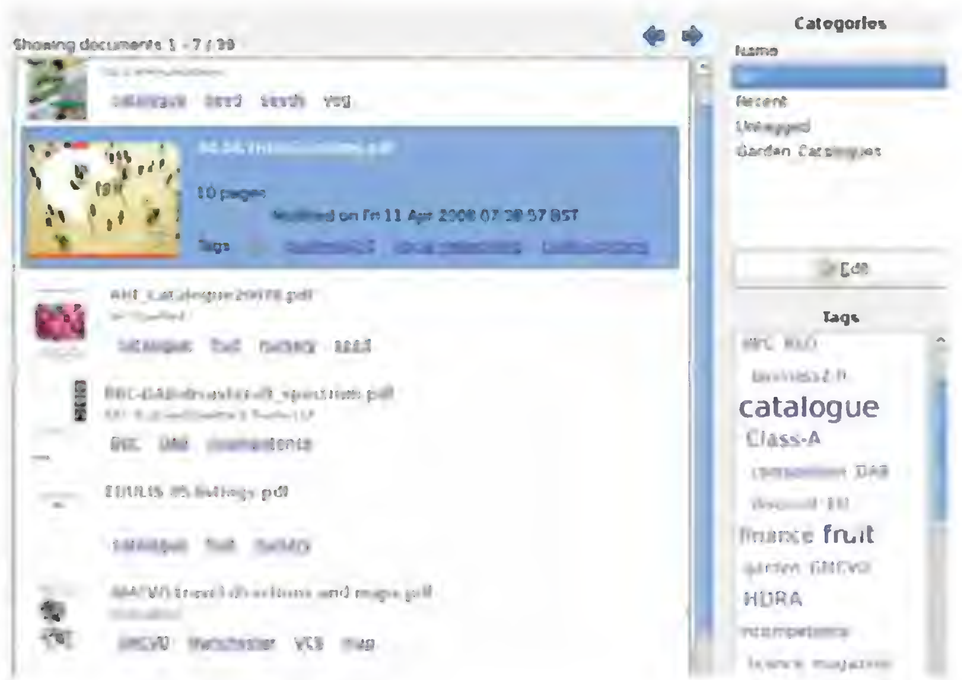
Приложение пока на ранней стадии развития, и хотя установка не особо трудна, о зависимостях надо заботиться самим. Если «верить на слово» менеджеру пакетов, то всё необходимое уже есть; но стадия **./configure** живо вас переубедит, и вы кинетесь искать добавочные коды для компиляции – как пришлось делать нам с новейшей версией **gtkmm-utils** на Ubuntu 8.04. Спасибо GNU *Make*, существенно облегчающей процесс: вскоре *PaperBox* красовался в нашем меню «Программы».

За учёт и контроль

Если *Tracker* не установлен на индексацию системы, ничего не получится. Поэтому запускайте *PaperBox* только после соответствующей настройки *Tracker*. Программа отобразит все документы компьютера, автоматически вычленив их из базы данных *Tracker*, и вам придется повозиться с этим массивом информации: время добавлять тэги!

Версия 0.2 получила новые MIME-типы: кое-что из OpenDocument, *AbiWord*, *Microsoft Office*, *PostScript* и простой текст. Теперь программа взаимодействует с *Tracker* асинхронно, и расстановка тэгов в *PaperBox* не помешает работе *Tracker*. А вот тэги, установленные инструментом индексации-поиска, сразу появятся в *PaperBox*.

Если у вас много электронных книг, PDF и других текстовых документов, *PaperBox* – более простой способ справиться с ними, по сравнению с нагромождением сложной структуры директорий. Кто не страшится борьбы с зависимостями – попробуйте.



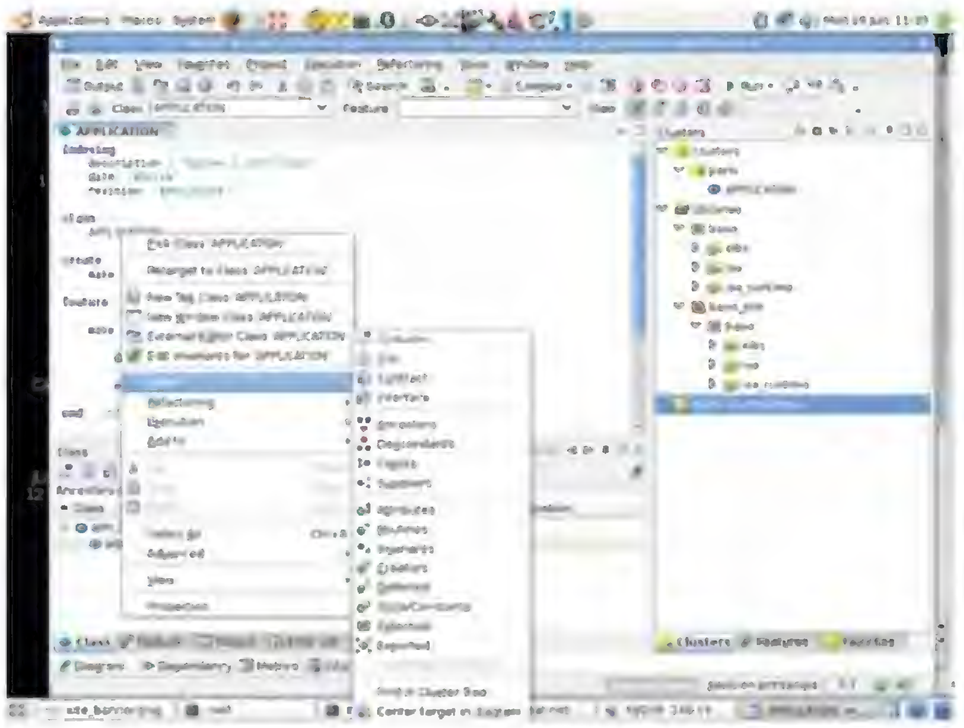
› *PaperBox* – удобный способ поиска документов за счёт облака тэгов и индексирования *Tracker*.

> «*PaperBox* — удобный браузер документов для Gnome.»

Объектно-ориентированная IDE

EiffelStudio

Версия 6.2.7.3753 **Сайт** www.eiffel.com/products/studio



› Мощный инспектор объектов отображает информацию о наследовании.

Те читатели **LXF**, что не работают программистами, как правило, делятся на две группы. В одной из них знают Perl и, возможно, Python, пользуясь ими для решения административных задач, и имеют некоторый опыт в C или C++, иногда Java. Другая группа изучает (или собирается изучать) один из 'P'-элементов LAMP – скриптовых языков типа Ruby или PHP. Обе группы могут выиграть от знакомства с Eiffel.

Почему так? А дело в том, что для улучшения навыков программирования нужны две вещи: чтение и написание кода на избранном языке и изучение хотя бы одного нового языка в год. Берясь за написание кода на новом языке, вы вынуждены решать задачи программирования новыми, неизвестными доселе способами. Свежее мышление, приобретённое «на стороне», очень помогает по возвращении в привычную рабочую среду; с этим согласны многие опытные программисты.

Eiffel Бертрана Мейера [Bertrand Meyer] – это чистый объектно-ориентированный (ОО) язык, подобный Smalltalk, без ощущения неуклюжести и излишеств, присущих ООП на Java или C++. Eiffel имеет организационные функции, например, кластеры (группы классов), что удобно для управления проектами, особенно в команде. В отличие от Smalltalk, типы здесь статические, и это вносит фактор личного вкуса: Eiffel или любят, или ненавидят. Но выгоду от его изучения, несомненно, получит любой программист, независимо от своих предпочтений.

Французский способ

Недавно Eiffel прошёл процесс стандартизации в ECMA (Европейской ассоциации производителей компьютеров, European Computer Manufacturers' Association) и ISO, что привело к появлению всяких нововведений, от команд-присваивателей и скобочной нотации до inline-агентов и нечёткого наследования. Компилятор *SmartEiffel*, предоставленный GNU-проекту Лабораторией теоретических и прикладных исследований в области информационных технологий (Loria, Лотарингия), отошёл от стандарта ECMA, сохранив прежний дух Eiffel. Если этот язык вам понравится, вероятно, вы захотите попробовать и *SmartEiffel*, и *EiffelStudio*, и лично выбрать созвучную вам IDE.

> «Eiffel Бертрана Мейера объектно-ориентирован, наподобие Smalltalk.»

Зависимости у *EiffelStudio* весьма консервативны. Имея любую версию GTK не старше трёх лет, считайте себя в дамках. Из аннотации ясно, что это несложная IDE со стандартным вкладочным интерфейсом редактирования, древом директорий и описаний, мощным инспектором объектов и элементами высокоуровневого дизайна. Компилятор следует парадигме «устойчивого отказа», характерной для Eiffel, и выявляет до 90% ошибок.

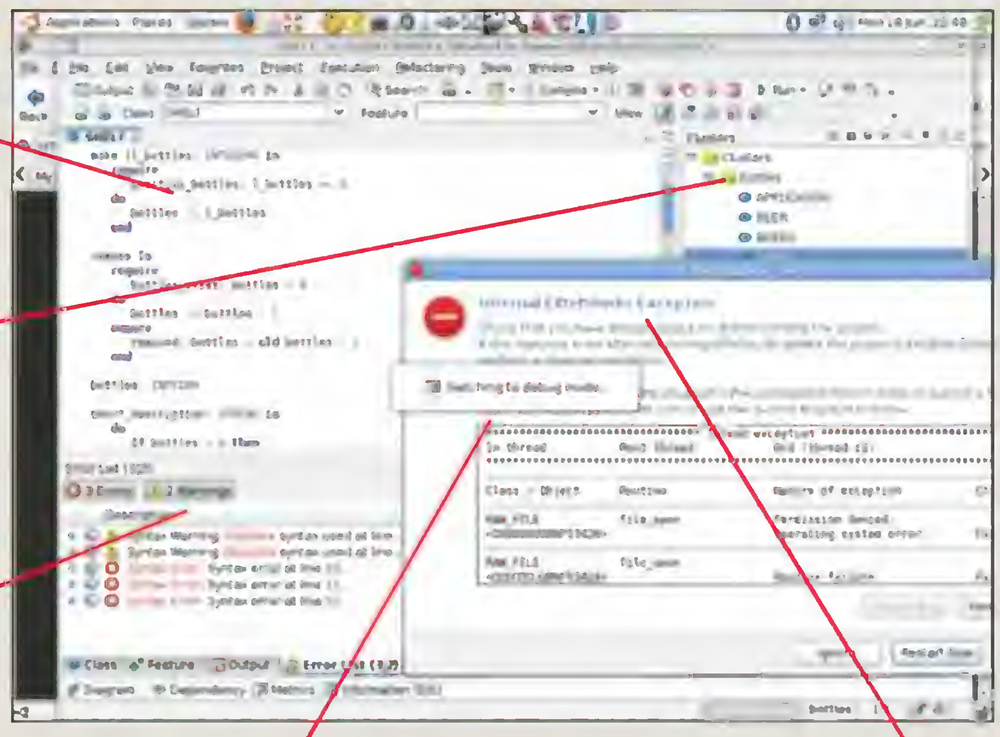
Добив все ошибки, компиляция выдает код либо на C, либо на .NET. Мы выбрали C и обнаружили, что код на выходе близок к оптимальному, и предусмотрена возможность ручной правки. Но для большинства пользователей качество от *EiffelStudio* более чем хорошее, причём высокоуровневый подход к проектированию и реализации часто исключает даже промежуточную UML-стадию в некоторых проектах.

Медиа-яблочко

Что ещё? Eiffel не так уж популярен, к тому же из-за вмешательства ISO возникло два различных диалекта; но есть практическое применение и для него. Основанный на Eiffel язык *Apple Media Language*, применяемый в Apple Media – одна из возможных точек приложения навыков.

EiffelStudio доступна на многих платформах, и включает библиотеки для web, GUI, данных, сетей, XML и других популярных областей программирования. Подход «устойчивого отказа» к компиляции определённо симпатичен многим, и во многом облегчает выявление ошибок. Как и Smalltalk, Eiffel будет интересен для изучения в качестве первого языка программирования, и в Интернете найдётся масса ресурсов в помощь новичку.

Исследуем интерфейс EiffelStudio



Вкладки редактора
Здесь есть всё необходимое: подсветка синтаксиса и завершение кода, плюс инспектор объектов.

Кластеры
Классы собраны в кластеры – скорее организационная, чем синтаксическая конструкция. Eiffel-программы – это просто коллекции классов.

Перечень ошибок
После компиляции в специальном окне отображаются все обнаруженные ошибки. В нашем случае это результат использования устаревшего Eiffel-кода.

Режим устранения ошибок
До 90% ошибок выявляются на этапе компиляции, когда исправить их проще.

Внутренняя ошибка
Разрушающие испытания: запуск неисправной программы во внутреннем интерпретаторе может вывести *EiffelStudio* из строя.

HotGames Развлекательные приложения

Клон Worms

Wormux

Версия 0.8 **Сайт** www.wormux.org

Концепция *Worms* была столь удачна – бездумная мультяшная драка с хитрыми зверюгами – что мы уже не раз описывали её клоны в LXFHotpicks; но кросс-платформенная *Wormux* гостит у нас впервые.

Авторы *Wormux* характеризуют своё детище как «праздничную массовую бойню», напрочь отказываясь принимать что-либо всерьёз – кроме того, что игра сделана профессионально и на этот раз снабжена даже всем необходимым для изготовления загрузочного игрового CD!

При запуске игры звучит традиционная аркадная музыка и предлагается выбор команды и поля битвы. Никогда раньше не играли в Worms? Знайте, что вы попадёте в яркий мультипликационный 2D-мир, населённый монстриками, с богатым арсеналом смертоносной артиллерии. Затем последует пошаговый выбор стреляющего и пальба из всех видов оружия с целью нанести максимальный урон противнику. Игра не затягивает так плотно, как стратегия, но детей на полчаса прикуёт.

Это первая стабильная версия с поддержкой сети. Есть и одиночный режим, в нём все противники именуются просто 'AI-stupid' («придурки с искуственныи интеллектом»). Толпу вокруг компьютера, может, и не соберёт, но вполне увлекательно. Интересная особенность *Wormux* – состав персонажей. Здесь можно стать лисом из *Firefox*, гну-GNU, пингвином Tux или талисманом *Konqueror*, PHP и десятка других FOSS-проектов.

› **Выберите оружие и персонаж и насладитесь «праздничной массовой бойней».**

> **«Вас ждут монстрики и богатый арсенал смертоносного оружия.»**

Если у вас всё ещё бета-версия, полученная с Ubuntu Hardy, обновляйтесь ради свежих карт, улучшения звука и нового вооружения. Если же вы до сих пор не знакомы с *Wormux* – пакеты есть для большинства стандартных дистрибутивов, пробуйте смело.

Головоломки

NPush

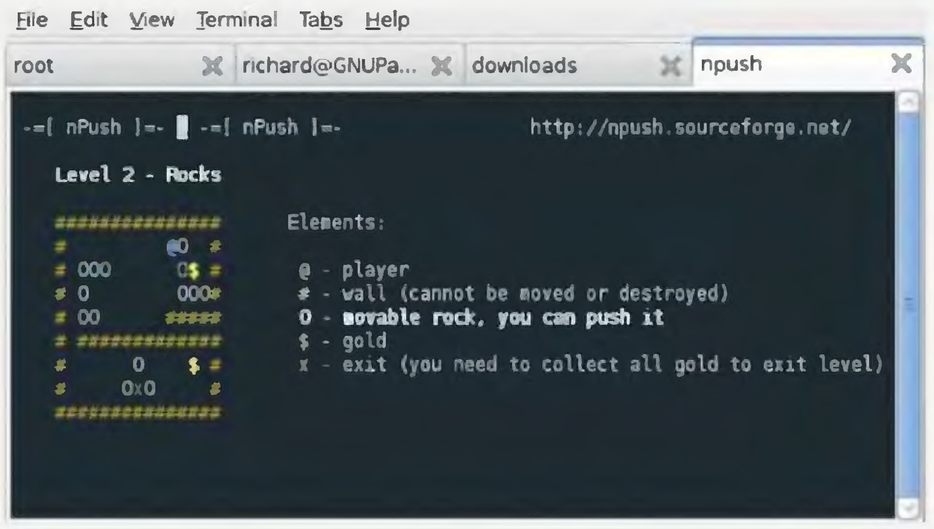
Версия 0.6 **Сайт** http://npush.sourceforge.net

Всем, кто «завязал» с игрой в *Sokoban*, советуем немедля бросить чтение – это ваш последний шанс удержаться от нового срыва. *Sokoban* – увлекательная игра японского происхождения, гулявшая по игровых приставкам всего Дальнего Востока в 80-х. В 90-х она пережила что-то вроде ренессанса, когда исследователи искусственного интеллекта занялись проблемой NP (для математиков: недетерминированного полиномиального времени, ха-ха).

В классической *Sokoban* игрок проталкивает ящики на свои места сквозь лабиринт геометрических фигур. Игра доступна для X11, KDE и, конечно, *Emacs*, и по-прежнему увлекательна. *NPush* взяла *Sokoban*, добавила *Boulder Dash* и чуть-чуть *Robots*, и получился коктейль для умственной гимнастики, тренирующий мозг не хуже 10-минутной развивающей игры на Nintendo.

Золотая лихорадка

Для завершения уровня вам нужно собрать все символы $ (обозначающие золото) в игровом пространстве, и найти выход. Если проход закрывают валуны, воспользуйтесь динамитом – только вот заложить его будет не так просто, придётся поразмыслить. Ещё одно отличие от *Sokoban* – несколько персонажей (между ними можно переключаться клавишей Tab), общими усилиями добивающихся одной цели. Интерфейс *NPush* написан с помощью знакомой библиотеки *Ncurses*, само приложение – на C++, но с расчётом на портирование. Поэтому если подгонять игру, скажем, к библиотеке SDL, в основном коде придётся менять не так уж много. Сам автор приглашает принять участие в кодировании всех желающих.



› **Двигайте валуны, но не заваливайте путь, соберите всё золото и найдите выход – скоро задача усложнится.**

> ***«NPush* — это коктейль для гимнастики ума, он держит разум в тонусе.»**

Хорошая новость – по крайней мере, для не-программистов: уровни – это просто текстовые файлы, и составить собственный совсем не трудно. А можно, пройдя все уровни, просто вернуться к своей *Sokoban*.

Редактор гитарных табулатур

TuxGuitar

Версия 1.0RC4 **Сайт** www.tuxguitar.com.ar

TuxGuitar – редактор гитарных табулатур и плейер, и он отлично подходит для большинства нужд гитаристов. Написанная на Java, программа работает и с GCJ, и с IcedTea, и с JVM от Sun. А кому не нужен редактор для написания музыки, тем придутся по душе одни только возможности воспроизведения *TuxGuitar*.

Поскольку *Guitar Pro* – лидер рынка (хотя программа недоступна на Linux), все остальные плейеры обречены на совместимость с богатой библиотекой гитарных аккордов, записанных в его формате. *TuxGuitar* совместим с файлами, созданными в трёх последних версиях *Guitar Pro*, широко распространёнными в Интернете. Напомним о необходимости соблюдения авторских прав на музыкальные произведения (мы тестировали программу с музыкой, на которую также имели ноты; аспекты права в разных странах различны).

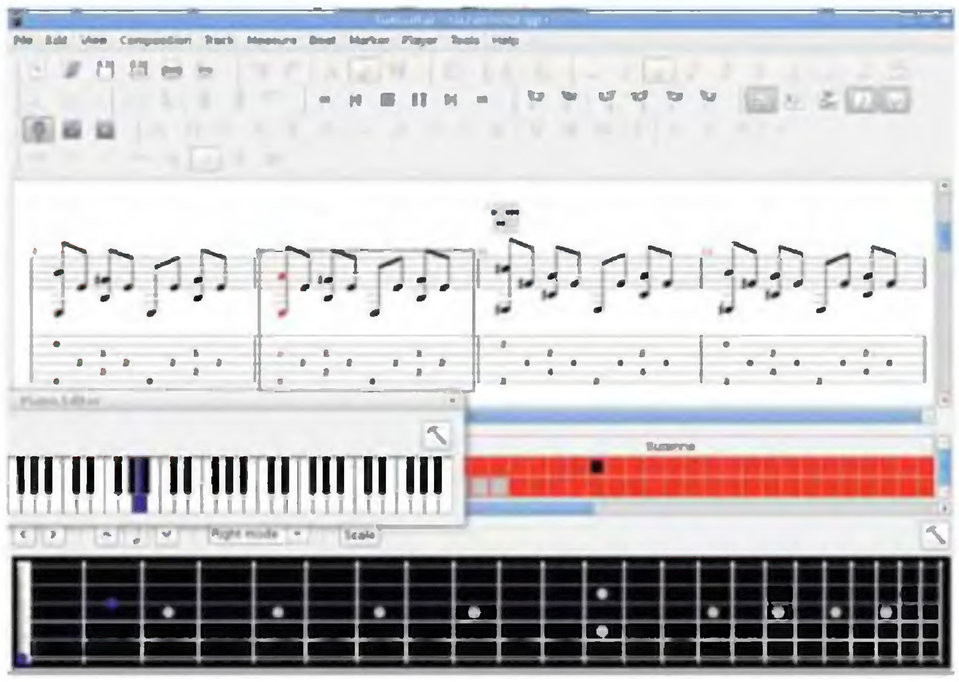
Редактор *TuxGuitar* предоставляет удобный клавиатурный ввод для написания табулатур, сопровождаемых множеством добавочных эффектов, вроде аккордов, взятых с медиатором, трелей и стука. Легко обрабатываются параметры времени и темпа. Онлайн-документация описывает редактор в мельчайших деталях, а новичков вкратце знакомит с музыкальной грамотой.

Прелести интерфейса

Хотя музыка хранится на ПК, её воспроизведение нужно композитору, чтобы прослушать заковыристый пассаж, а музыканту – чтобы выучить новую пьесу. По умолчанию в окне программы отображается нотная запись и партитура, причём по ходу игры ноты подсвечиваются. При желании можно вывести клавиатуру или гриф гитары (см. экранный снимок).

Возможно воспроизведение других инструментов (например, гармоники), причём для переложения пьесы для гитары или с целью импровизации звучание может дублироваться на экранном грифе.

> «*TuxGuitar* стоит признания уже за его воспроизведение.»



› **Показ клавиатуры и грифа позволяют быстро «схватить» мелодию новой пьесы и играть её по нотам.**

В **LXF104** мы рассматривали *KGuitar* и высоко оценили его минималистский удобный дизайн, интеграцию с KDE и наличие импорта/экспорта MIDI и MusiXML. *TuxGuitar* добавляет совместимость с *Guitar Pro* 5, экспорт в *LilyPond* и множество полезных функций редактирования, не теряя ясности интерфейса. Явный победитель.

Загрузочный диск Linux

BG-Tiny Linux Bootdisk

Версия 0.8 **Сайт** www.giannone.eu/bgtlb/current

Все мы порой пользуемся «спасательными» дисками. Для восстановления повреждённого ПК обычно вполне хватает загрузочного CD Knoppix или Ubuntu. Но у старых машин или ноутбуков слишком мало ОЗУ для таких «тяжеловесов»: тут нужны инструменты полегче. А легче BG-Tiny Linux Bootdisk даже и не найти.

BG-Tiny – это загрузочный диск, умещающийся на дискете, вот до чего он мал. Да ещё ему не нужен RAM-диск, поэтому с него можно загружать PC с 386 процессором и всего 4 МБ ОЗУ. Порывшись в куче ноутбуков на антресолях **LXF**, мы обнаружили стопку IBM ThinkPad 701 с объёмом оперативной памяти от 4 до 24 МБ каждый, чего едва хватало предустановленным OS/2. Мы подключили внешний дисковод к тому, на котором ещё теплилась батарея CMOS, и – вуаля – Linux заработал!

Видеть ожившую машину родом из 90-х, конечно, любопытно, но где тут польза? Случается, на старых испорченных компьютерах остаются ценные файлы или диски, которые можно поправить с помощью *fsck*. BG-Tiny даёт для этого немногим больше, чем базовый набор программ, включая *BusyBox*, но в большинстве случаев этого и достаточно.

Оставшаяся память распределяется между *dosfsprogs*, *e2fsprogs*, *Gpart*, *Lilo* и *ms-sys*, втиснутой для работы с файловыми системами MS.

Предусмотрено тридцать клавиатурных раскладок – поддерживаются даже USB-клавиатуры. Ядро Linux 2.4.36 на этом узеньком пространстве памяти как раз по двору, и BG-Tiny оставляет вам аж 844 КБ ОЗУ на 4-мегабайтной машине. Судя по тому, что версия 0.7 оставляла 772 КБ, а версия 0.6 – 672 КБ, автор диска Бодо Джанноне [Bodo Giannone] от души сочувствует работающим с компьютерными древностями.

> «*BG-Tiny* загрузится на ПК с 386 процессором и всего 4 МБ ОЗУ.»

› **Привет из прошлого – дряхлый аппарат поверяет свои пыльные секреты BG-Tiny Linux Bootdisk.**

Если вы занимаетесь старым «железом», дискета вполне заслуживает места в вашем портфеле рядом с набором отвёрток Torx, диском SGD и PDF «*DOS для чайников*».

Органайзер

YAGTD

Версия 0.2.4 **Сайт** https://gna.org/projects/yagtd

С учётом количества статей об использовании организационного и управленческого ПО, не говоря уже о широте распространения приложений-органайзеров, просто удивительно, что компьютерщики ещё не завладели миром. Или завладели, да никто не заметил?

Если вы склоняетесь к GTD (Getting Things Done, Как привести дела в порядок) – методологии Дэвида Аллена [David Allen], то *YAGTD* (Yet Another Getting Things Done, Еще один способ привести дела в порядок) создан для вас.

На несколько порядков легче *Chandler* (**LXF107**) или *Org-mode* редактора *Emacs*, легче даже *Tomboy* для Gnome, *YAGTD* – это текстовое приложение для слежения за вашими планами. Многое в программе взято из книги *Seven Habits of Highly Effective People* [Семь правил успешных людей] Стивена Кови [Steven Covey]. Причём это не сатрап, явившийся на ваш ПК наводить свои порядки, а просто список намеченных дел с проставлением приоритетов и важности.

```
File  Edit  View  Terminal  Tabs  Help
root | richard@GNUPa... | downloads | yaGTD
79 tasks found
GTD> help

Documented commands (type help <topic>):
========================================
EOF         del         la       ls         recurrence  show      time
add         done        list     maybe      ref         someday   urgency
append      due         listall  modify     replace     sort      waitingfor
appendall   edit        listpri  modifyall  rm          start
archive     end         listref  notes      save        status
close       extend      load     order      search      sub
complete    importance  lr       print      searchall   summary

Undocumented commands:
======================
help

GTD> listpri
  1: Next actions - A thousand mile journey begins with one step...
  2: * Calls
  3: ** TV Aerial -
  4: ** Steve Durrant, Manchester Green Party - 07[illegible]
```

Имя у *YAGTD* неблагозвучное, но принципы здравые.

Возьмите с собой USB-накопитель с программой (**python yadtd.py** позволяет работать на любой платформе с поддержкой Python) – и вы на пути к организационному совершенству. В пакет включён модуль для *Emacs*, **yagtd-mode.el**.

Облегчённый текстовый редактор

Zile

Версия 2.2.59 **Сайт** www.gnu.org/software/zile

LINUX FORMAT HotPicks Revisited

Zile – это попросту урезанная версия *Emacs*, задуманная для базового редактирования текста в условиях ограничений по памяти (на спасательной дискете или встраиваемой системе).

Многие клавиатурные комбинации и названия функций аналогичны *Emacs*, и поклонники величайшего из редакторов изучат *Zile* без особого напряга. Но функции *Zile* далеко не ограничиваются базовым набором. Например, программа умеет отображать несколько окон редактирования в терминале – отнюдь не характерно для столь компактных приложений. Возможность редактирования нескольких файлов ограничена лишь размерами оперативной памяти, а серией откатов можно восстановить любое состояние документа вплоть до начала сеанса.

Как и в *Emacs*, команды набираются в мини-буфере; для самих команд, названий файлов и переменных возможно автозавершение. В предельно ограниченном пространстве *Zile* разворачивает бурную деятельность, которая не по зубам сходным по размерам редакторам.

Опытные пользователи *Emacs* осудят нехватку части клавиатурных комбинаций, а поклонников *Vi* оттолкнут выкрутасы типа Ctrl+Meta+Alt+Shift, обычные для сеансов *Emacs*. И всё же в *Zile*, как в сжатую пружину, удалось вложить большую часть энергии «старшего брата». LXF

Почти *Emacs*, но найти десять отличий при желании можно.

```
File  Edit  View  Terminal  Tabs  Help
Thu 2008/04/03 22:12:06: Escaping SMS mode
Thu 2008/04/03 22:12:06: Sending simple AT command to wake up som$
Thu 2008/04/03 22:12:06: 1 "AT"
Thu 2008/04/03 22:12:06: 2 "OK"
Thu 2008/04/03 22:12:06: Enabling echo
Thu 2008/04/03 22:12:07: 1 "ATE1"
Thu 2008/04/03 22:12:07: 2 "OK"
Thu 2008/04/03 22:12:07: Trying Motorola mode switch
Thu 2008/04/03 22:12:07: 1 "AT+MODE=2"
Thu 2008/04/03 22:12:07: 2 "ERROR"
Thu 2008/04/03 22:12:07: Seems not to be supported
Thu 2008/04/03 22:12:07: Enabling CME errors
Thu 2008/04/03 22:12:08: 1 "AT+CMEE=1"
Thu 2008/04/03 22:12:08: 2 "OK"
Thu 2008/04/03 22:12:08: 1 "AT+CSCS?"
Thu 2008/04/03 22:12:08: 2 "+CSCS: "GSM""
Thu 2008/04/03 22:12:08: 3 "OK"
Thu 2008/04/03 22:12:08: 1 "AT+CGMI"
Thu 2008/04/03 22:12:08: 2 "Sony Ericsson"
Thu 2008/04/03 22:12:08: 3 "OK"
Thu 2008/04/03 22:12:08: Manufacturer info received
Thu 2008/04/03 22:12:08: [Manufacturer: Sony Ericsson]
--:--  wammu.log          1% (39,0)    (Text)----------
M-p is undefined
```

Также вышли...

Новые и обновленные программы, на которые тоже стоит обратить внимание...

Sendmail 8.14.3
Почтенный почтовый сервер прошел новый этап исправлений.
www.sendmail.org

Cosmo 0.14.2
Усовершенствован календарный OSAF-сервер Chandler.
http://chandlerproject.org/server

Wiki2beamer 0.5
Создание LaTeX-презентаций с использованием синтаксиса wiki.
http://wiki2beamer.sourceforge.net

Security Management and Risk Tracking (Smart) 3.0.0
Онлайн-комплекс управления информационной безопасностью.
http://smart.conformix.com

SuperTuxKart 0.5
Новые трассы и функции.
http://supertuxkart.sourceforge.net

Kite 1.0b1
«Плавающий язык программирования». Лёгкий, несложный и скорый в разработке.
www.kite-language.org

Oroboros 20080505
Астрологическое приложение, основанное на швейцарских таблицах эфемерид, базе данных временных зон Ольсона, комплекте *Qt 4* и тарабарском жаргоне.
http://pypi.python.org/pypi/oroboros

StudioDix 0.1
Java-ПО для управления студией звукозаписи.
http://studiodix.wiki.sourceforge.net

LINUX FORMAT
Главное в мире Linux

Журнал зарегистрирован Федеральной службой по надзору за соблюдением законодательства в сфере массовых коммуникаций и охране культурного наследия
ПИ № ФС77-21973 от 14 сентября 2005 года
Выходит ежемесячно. Тираж 6000 экз.

РЕДАКЦИЯ РУССКОЯЗЫЧНОЙ ВЕРСИИ:
ГЛАВНЫЙ РЕДАКТОР
Валентин Синицын info@linuxformat.ru
ВЫПУСКАЮЩИЙ РЕДАКТОР
Родион Водейко
Литературные редакторы
Елена Толстякова
Переводчики
Илья Аввакумов, Александр Бикмеев, Светлана Кривошеина, Александр Казанцев, Алексей Опарин, Валентин Развозжаев
Редактор диска
Александр Кузьменков
Допечатная подготовка
Мария Пучкова, Родион Водейко, Семен Фрумкин
Креативный директор
Станислав Медведев
Технический директор
Денис Филиппов
Директор по рекламе
Денис Игнатов +7 812 965 7236 advert@linuxformat.ru
Генеральный директор
Павел Фролов
Заместитель генерального директора
Софья Виниченко
УЧРЕДИТЕЛИ
частные лица
ИЗДАТЕЛИ
Станислав Медведев, Виктор Федосеев, Павел Фролов

Отпечатано в типографии «Текст», ООО «ППК «Текст»
188680, Ленинградская область, Всеволожский район, Колтуши, д.32
Заказ _____

Пре-пресс: *d.r.i.v.e-group*

РЕДАКЦИЯ АНГЛОЯЗЫЧНОЙ ВЕРСИИ:
Редактор Пол Хадсон [Paul Hudson] paul.hudson@futurenet.co.uk
Редактор обзоров Грэм Моррисон [Graham Morrison]
Редактор новостей Майк Сондерс [Mike Saunders] mike.saunders@futurenet.co.uk
Художественный редактор Эфрейн Эрнандес-Мендоса (Efrain Hernandez-Mendoza) efrain.hernandez-mendoza@futurenet.co.uk
Литературный редактор Эндрью Грегори [Andrew Gregory] agregory@futurenet.co.uk

Подготовка материалов
Нейл Ботвик [Neil Bothwick], д-р Крис Браун [Dr. Chris Brown], Энди Ченнел [Andy Channelle], Майкл Дж. Хэммел [Michael J. Hammel], Бен Харлинг [Ben Harling], Дэниел Джеймс [Daniel James], Сьюзен Линтон [Susan Linton], Джеймс Литтон [James Litton], Саймон Пикстоск [Simon Pickstock], Ричард Смедли [Richard Smedley], Ник Вейч [Nick Veitch], Уилл Трибби [Will Tribbey], Алекс Янг [Alex Young], Евгений Балдин, Александр Бикмеев, Андрей Боровский, Антон Коробейников, Евгений Крестников, Алексей Маслий, Галина Пожарина, Андрей Прахов, Алексей Федорчук, Виктор Федосеев
Художественные ассистенты: Стейси Блек [Stacey Black], Фил Хэйкрэфт [Phil Haycraft], Кейт МакДоннелл [Kate McDonnell]
Иллюстрации: Elly Walton Illustrations, Грэм Моррисон [Graham Morrison], Крис Винн [Chris Winn], iStock Photo

КОНТАКТНАЯ ИНФОРМАЦИЯ
UK: Linux Format, 30 Monmouth Street, Bath BA1 2BW
Tel 01225 442244 Email: linuxformat@futurenet.co.uk
РОССИЯ:
Санкт-Петербург (редакция): Лиговский пр. 50 корп. 15, тел. +7 (812) 640-49-90
Представительство в Москве:
ул. Энергетическая, д.14, корпус 5, стр. 1
По вопросам сотрудничества, партнерства, оптовых закупок: +7 (495) 799-18-63
Дирекция московского офиса: +7 (495) 136-88-45. E-mail: moscow@linuxcenter.ru

Авторские права: Статьи, переведенные из английского издания Linux Format, являются собственностью или лицензией Future Publishing Ltd (Future plc group company). Все права зарегистрированы. Никакая часть данного журнала не может быть повторно опубликована без письменного разрешения издателя.

Все письма, независимо от способа отправки, считаются предназначенными для публикации, если иное не указано явно. Редакция оставляет за собой право корректировать присланные письма и другие материалы. Редакция Linux Format получает неэксклюзивное право на публикацию и лицензирование всех присланных материалов, если не было оговорено иное. Linux Format стремится оставлять уведомление об авторских правах всюду, где это возможно. Свяжитесь с нами, если мы не упомянули вас как автора предложенных вами материалов и мы постараемся исправить эту ошибку. Редакция Linux Format не несет ответственности за опечатки.

Все присланные материалы могут быть помещены на CD или DVD-диски, поставляемые вместе с журналом, если не было оговорено иное.

Ограничение ответственности: используйте все советы на свой страх и риск. Ни при каких условиях редакция Linux Format не несет ответственность за повреждения или ущерб, нанесенные вашему компьютеру и переферии вследствие использования тех или иных советов.

LINUX – зарегистрированный товарный знак Линуса Торвальдса [Linux Torvalds]. «GNU/Linux» заменяется на «Linux» в целях сокращения. Все остальные товарные знаки являются собственностью их законных владельцев. Весь код, опубликованный в журнале, лицензирован на условиях GPL v3. См. www.gnu.org/copyleft/gpl.html.

Ответственность за содержание статьи несет ее автор. Мнение авторов может не совпадать с мнением редакции.

За информацией о журналах, издаваемых Future plc group company, обращайтесь http://www.futureplc.com

© Linux Format 2005

© Future Publishing Ltd 2005

в октябре

Лучшее из мира свободного ПО

От web-браузеров до текстовых редакторов: мы найдем все лучшее, что может предложить нам Open Source. Пользователи Linux – прогревайте свои пакетные менеджеры.

Учиться и платить?

Курсы по Linux – дорогостоящая трата времени или прекрасный способ извлечь выгоду из ваших навыков? Попробуем разобраться...

Советы по мультимедиа

Продайте свой Macintosh – в Linux можно делать все то же самое.

MINIX 3

Все пользователи Linux слышали о нем, но не все представляют себе, как он выглядит: мы восполним этот пробел.

Содержание последующих выпусков может меняться без уведомления

Future
BATH • LONDON • MILAN • NEW YORK • PARIS • SAN DIEGO •SAN FRANCISCO